# class1绚丽的时钟效果

Canvas提供的函数

Canvas.width

Canvas.height

Canvas.getContext(‘2d’)

## canvas绘制基础

## 创建canvas

兼容性

<canvas id="canvas" style="border:1px solid #aaa;display:block;margin:50 auto;">

当前浏览器不支持Canvas

</canvas>

<script>

if(canvas.getContext('2d')){

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

}else{

alert('当前浏览器不支持Canvas');

}

</script>

### 绘制直线、多边形

基于状态的绘图（先设定状态，后调用函数进行具体的绘制）

绘制直线

context.moveTo(100,100);

context.lineTo(700,700);

context.lineTo(100,700);

context.lineTo(100,100);

context.lineWidth = 5;

context.strokeStyle = "#005588";

context.stroke();

context.moveTo(100,100);

context.lineTo(700,700);

context.lineTo(100,700);

context.lineTo(100,100);

context.fillStyle = "rgb(2,100,300)";

context.fill();

context.lineWidth = 5;

context.strokeStyle = "red";

context.stroke();

基于状态的

context.moveTo(100,100);

context.lineTo(700,700);

context.lineTo(100,700);

context.lineTo(100,100);

context.lineWidth = 5;

context.strokeStyle = "red";

context.stroke();

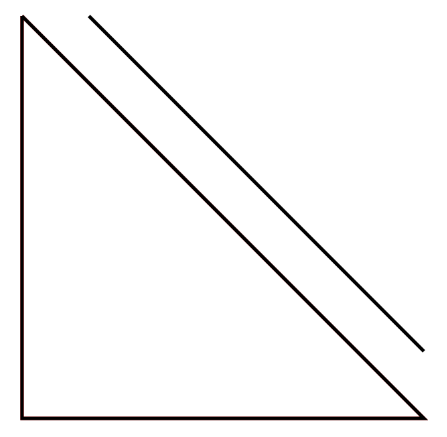
context.moveTo(200,100);

context.lineTo(700,600);

context.strokeStyle = "black";

context.stroke();

两段路径的颜色和宽度相同



多个路径分开处理

context.beginPath()

context.closePath

碰撞检测

实例：绘制七巧板

var tangram =[

{p:[{x:0,y:0},{x:800,y:0},{x:400,y:400}],color:"#caff67"},

{p:[{x:0,y:0},{x:400,y:400},{x:0,y:800}],color:"#67becf"},

{p:[{x:800,y:0},{x:800,y:400},{x:600,y:600},{x:600,y:200}],color:"#ef3d61"},

{p:[{x:600,y:200},{x:600,y:600},{x:400,y:400}],color:"#f9f51a"},

{p:[{x:400,y:400},{x:600,y:600},{x:400,y:800},{x:200,y:600}],color:"#a594c0"},

{p:[{x:200,y:600},{x:400,y:800},{x:0,y:800}],color:"#fa8ccc"},

{p:[{x:800,y:400},{x:800,y:800},{x:400,y:800}],color:"#f6ca29"}

]

window.onload = function(){

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

canvas.width = 800;

canvas.height = 800;

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

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

draw(tangram[i],context);

}

function draw (piece ,cxt) {

cxt.beginPath();

cxt.moveTo(piece.p[0].x,piece.p[0].y);

for(var i =1;i<piece.p.length;i++)

cxt.lineTo(piece.p[i].x,piece.p[i].y);

cxt.closePath();

cxt.fillStyle = piece.color;

cxt.fill();

cxt.strokeStyle = "black";

cxt.lineWidth = 3;

cxt.stroke();

}

}

### 绘制弧线和圆

contex.arc(

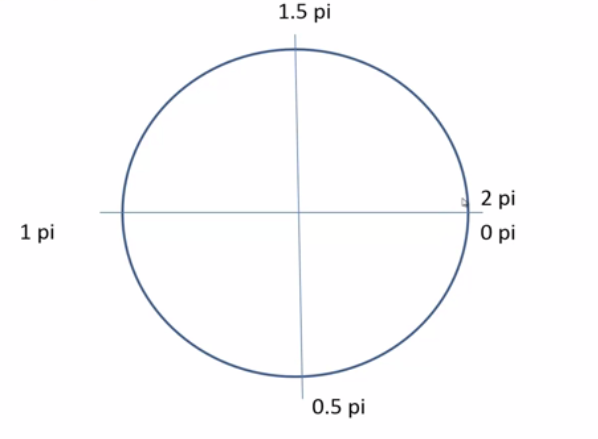
centerx,centery,radius,

startingAngle,endingAngle,

anticlochwise=false

)

无论是逆时针还是瞬时针，度数不变，如下：



var canvas = document.getElementById("canvas");

canvas.width=1024;

canvas.height=768;

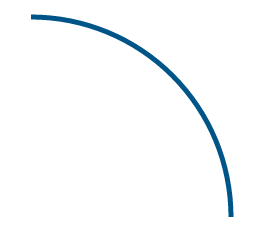
var context = canvas.getContext('2d');

context.lineWidth =5;

context.strokeStyle = "#005588";

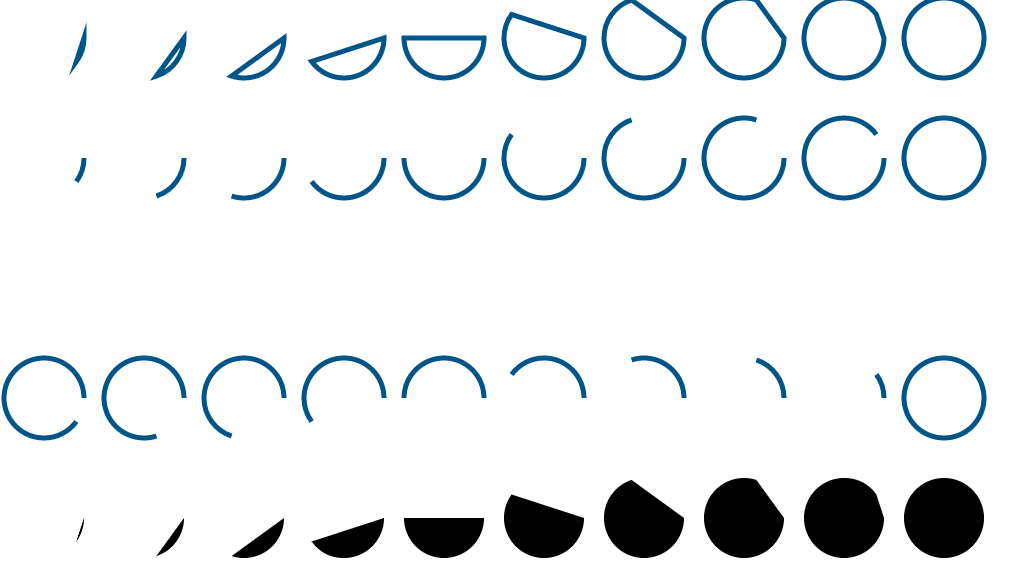
context.arc(300,300,200,0,1.5\*Math.PI,true);

context.stroke();



closePath:如果绘制的图形没有闭合，使用closePath会自动闭合所绘图形

beginPath和closePath不一定要重复出现



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

context.beginPath();

context.arc(50+i\*100,60,40,0,2\*Math.PI\*(i+1)/10);

context.closePath();

context.stroke();

}

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

context.beginPath();

context.arc(50+i\*100,180,40,0,2\*Math.PI\*(i+1)/10);

context.stroke();

}

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

context.beginPath();

context.arc(50+i\*100,420,40,0,2\*Math.PI\*(i+1)/10,true);

context.stroke();

}

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

context.beginPath();

context.arc(50+i\*100,540,40,0,2\*Math.PI\*(i+1)/10);

context.closePath();

context.fill();

}

如果没有closePath，就直接使用fill，调用fill的时候，canvas会把没有封闭的路径首尾相连

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

context.beginPath();

context.arc(50+i\*100,730,40,0,2\*Math.PI\*(i+1)/10);

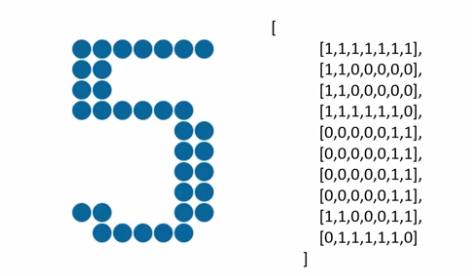
context.fill();

}

## 倒计时电子钟的实现

### 倒计时程序的基本架构

二维点阵模型



window.onload = function(){

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

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

canvas.width = WINDOW\_WINTH;

canvas.height = WINDOW\_HEIGHT;

render(context);

}

function render(cxt){

var hours =12;

var minutes =34;

var seconds =56;

renderDigit(0,0,parseInt(hours/10),cxt);//一定要传绘图上下文环境

}

function renderDigit(x,y,num,cxt){

cxt.fillStyle = "rgb(0,102,153)";

for(var i=0;i<digit[num].length;i++){

for(var j=0;j<digit[num][i].length;j++){

if(digit[num][i][j]==1){

cxt.beginPath();

cxt.arc();

cxt.closePath();

cxt.fill();

}

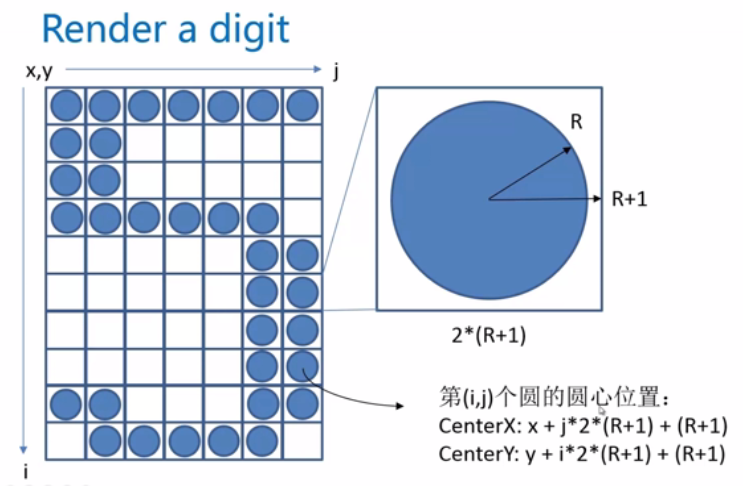
}

}

}

### 倒计时数字中的具体绘制

圆心位置的确定【格子系统】



### 倒计时的时间计算

不用写死的时间

## 绚丽的动画效果

### 一个实现动画的基础模式



### 使用canvas做一个物理实验

<!DOCTYPE html>

<html>

<head>

<title></title>

</head>

<body>

<canvas id="canvas" style="border:1px solid #aaa;display:block;margin:50 auto;">

当前浏览器不支持Canvas

</canvas>

<script type="text/javascript">

var ball={x:512,y:100,r:20,g:2,vx:-4,vy:-10,color:"#005588"}

window.onload=function(){

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

canvas.width = 1024;

canvas.height = 678;

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

setInterval(function(){

render(context);

update();

},50);

}

function update(){

ball.x+=ball.vx;

ball.y+=ball.vy;

ball.vy+=ball.g;

}

function render(cxt){

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

cxt.fillStyle = ball.color;

cxt.beginPath();

cxt.arc(ball.x,ball.y,10,0,2\*Math.PI);

cxt.closePath();

cxt.fill();

}

</script>

小球掉出了屏幕

加入“碰撞检测”和”空气阻力“

<!DOCTYPE html>

<html>

<head>

<title></title>

</head>

<body>

<canvas id="canvas" style="border:1px solid #aaa;display:block;margin:50 auto;">

当前浏览器不支持Canvas

</canvas>

<script type="text/javascript">

var ball={x:512,y:100,r:20,g:2,vx:-4,vy:-10,color:"#005588"}

window.onload=function(){

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

canvas.width = 1024;

canvas.height = 768;

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

setInterval(function(){

render(context);

update();

},50);

}

function update(){

ball.x+=ball.vx;

ball.y+=ball.vy;

ball.vy+=ball.g;

if(ball.y>=768-ball.r){

ball.y = 768 -ball.r;

ball.vy=-ball.vy\*0.5;

}

}

function render(cxt){

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

cxt.fillStyle = ball.color;

cxt.beginPath();

cxt.arc(ball.x,ball.y,ball.r,0,2\*Math.PI);

cxt.closePath();

cxt.fill();

}

</script>

### 华丽的小球滚动效果

<!DOCTYPE html>

<html>

<head>

<title></title>

</head>

<body>

<canvas id="canvas" style="border:1px solid #aaa;display:block;margin:50 auto;">

当前浏览器不支持Canvas

</canvas>

<script type="text/javascript">

var WINDOW\_WIDTH =1024;

var WINDOW\_HEIGHT = 768;

var RADIUS =8;

var MARGIN\_TOP=60;

var MARGIN\_LEFT=30;

const endTime = new Date(2016,3,12,18,47,52);//设置截止日期2014-07

var curShowTimeSeconds = 0;

var balls = [];//生成小球

var colors = ["#33B5E5","#0099CC","#AA66CC","#9933CC","#99CC00","#669900","#FFBB33","#FF8800","#FF4444","#CC0000"];

digit =

[

[

[0,0,1,1,1,0,0],

[0,1,1,0,1,1,0],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,0,1,1,0],

[0,0,1,1,1,0,0]

],//0

[

[0,0,0,1,1,0,0],

[0,1,1,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[1,1,1,1,1,1,1]

],//1

[

[0,1,1,1,1,1,0],

[1,1,0,0,0,1,1],

[0,0,0,0,0,1,1],

[0,0,0,0,1,1,0],

[0,0,0,1,1,0,0],

[0,0,1,1,0,0,0],

[0,1,1,0,0,0,0],

[1,1,0,0,0,0,0],

[1,1,0,0,0,1,1],

[1,1,1,1,1,1,1]

],//2

[

[1,1,1,1,1,1,1],

[0,0,0,0,0,1,1],

[0,0,0,0,1,1,0],

[0,0,0,1,1,0,0],

[0,0,1,1,1,0,0],

[0,0,0,0,1,1,0],

[0,0,0,0,0,1,1],

[0,0,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,1,1,1,0]

],//3

[

[0,0,0,0,1,1,0],

[0,0,0,1,1,1,0],

[0,0,1,1,1,1,0],

[0,1,1,0,1,1,0],

[1,1,0,0,1,1,0],

[1,1,1,1,1,1,1],

[0,0,0,0,1,1,0],

[0,0,0,0,1,1,0],

[0,0,0,0,1,1,0],

[0,0,0,1,1,1,1]

],//4

[

[1,1,1,1,1,1,1],

[1,1,0,0,0,0,0],

[1,1,0,0,0,0,0],

[1,1,1,1,1,1,0],

[0,0,0,0,0,1,1],

[0,0,0,0,0,1,1],

[0,0,0,0,0,1,1],

[0,0,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,1,1,1,0]

],//5

[

[0,0,0,0,1,1,0],

[0,0,1,1,0,0,0],

[0,1,1,0,0,0,0],

[1,1,0,0,0,0,0],

[1,1,0,1,1,1,0],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,1,1,1,0]

],//6

[

[1,1,1,1,1,1,1],

[1,1,0,0,0,1,1],

[0,0,0,0,1,1,0],

[0,0,0,0,1,1,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,1,1,0,0,0],

[0,0,1,1,0,0,0],

[0,0,1,1,0,0,0],

[0,0,1,1,0,0,0]

],//7

[

[0,1,1,1,1,1,0],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,1,1,1,0],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,1,1,1,0]

],//8

[

[0,1,1,1,1,1,0],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,1,0,1,1],

[0,0,0,0,0,1,1],

[0,0,0,0,0,1,1],

[0,0,0,0,1,1,0],

[0,0,0,1,1,0,0],

[0,1,1,0,0,0,0]

],//9

[

[0,0,0,0],

[0,0,0,0],

[0,1,1,0],

[0,1,1,0],

[0,0,0,0],

[0,0,0,0],

[0,1,1,0],

[0,1,1,0],

[0,0,0,0],

[0,0,0,0]

]//:

];

window.onload = function(){

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

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

canvas.width = WINDOW\_WIDTH;

canvas.height = WINDOW\_HEIGHT;

curShowTimeSeconds = getCurrentShowTimeSeconds();

setInterval(function(){

render(context);

update();//对当前数据进行调整

},50)

}

function getCurrentShowTimeSeconds(){

var curTime = new Date();

var ret =endTime.getTime() - curTime.getTime();

ret = Math.round(ret/1000);

return ret>0?ret:0;

}

function update(){

var nextShowTimeSeconds = getCurrentShowTimeSeconds();

var nextHours = parseInt(nextShowTimeSeconds/3600);

var nextMinutes =parseInt((nextShowTimeSeconds-nextHours\*3600)/60);

var nextSeconds =parseInt(nextShowTimeSeconds%60);

var curHours =parseInt(curShowTimeSeconds/3600);

var curMinutes =parseInt((curShowTimeSeconds-curHours\*3600)/60);

var curSeconds =parseInt(curShowTimeSeconds%60);

if(nextSeconds!=curSeconds){

if(parseInt(curHours/10)!=parseInt(nextHours/10)){

addBalls(MARGIN\_LEFT+0,MARGIN\_TOP,parseInt(curHours/10));

}

if(parseInt(curHours%10)!=parseInt(nextHours%10)){

addBalls(MARGIN\_LEFT+15\*(RADIUS+1),MARGIN\_TOP,parseInt(curHours%10));

}

if(parseInt(curMinutes/10)!=parseInt(nextMinutes/10)){

addBalls(MARGIN\_LEFT+39\*(RADIUS+1),MARGIN\_TOP,parseInt(curMinutes/10));

}

if(parseInt(curMinutes%10)!=parseInt(nextMinutes%10)){

addBalls(MARGIN\_LEFT+54\*(RADIUS+1),MARGIN\_TOP,parseInt(curMinutes%10));

}

if(parseInt(curSeconds/10)!=parseInt(nextSeconds/10)){

addBalls(MARGIN\_LEFT+78\*(RADIUS+1),MARGIN\_TOP,parseInt(curSeconds/10));

}

if(parseInt(curSeconds%10)!=parseInt(nextSeconds%10)){

addBalls(MARGIN\_LEFT+93\*(RADIUS+1),MARGIN\_TOP,parseInt(curSeconds%10));

}

curShowTimeSeconds = nextShowTimeSeconds;

}

updateBalls();

}

function updateBalls () {

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

balls[i].x+=balls[i].vx;

balls[i].y+=balls[i].vy;

balls[i].vy+=balls[i].g;

if(balls[i].y>=WINDOW\_HEIGHT-RADIUS){

balls[i].y=WINDOW\_HEIGHT-RADIUS;

balls[i].vy = -balls[i].vy\*0.75;

}

}

}

function addBalls(x,y,num){

for(var i=0;i<digit[num].length;i++){

for(var j=0;j<digit[num][i].length;j++){

if(digit[num][i][j]==1){

var aBall={

x:x+j\*2\*(RADIUS+1)+(RADIUS+1),

y:y+i\*2\*(RADIUS+1)+(RADIUS+1),

g:1.5+Math.random(),

vx:Math.pow(-1,Math.ceil(Math.random()\*1000))\*4,

vy:-5,

colors:colors[Math.floor(Math.random()\*colors.length)]

}

balls.push(aBall);

}

}

}

}

function render(cxt){

cxt.clearRect(0,0,WINDOW\_WIDTH,WINDOW\_HEIGHT);//清空屏幕

var hours =parseInt(curShowTimeSeconds/3600);

var minutes =parseInt((curShowTimeSeconds-hours\*3600)/60);

var seconds =parseInt(curShowTimeSeconds%60);

renderDigit(MARGIN\_LEFT,MARGIN\_TOP,parseInt(hours/10),cxt);//一定要传绘图上下文环境

renderDigit(MARGIN\_LEFT+15\*(RADIUS+1),MARGIN\_TOP,parseInt(hours%10),cxt);

renderDigit(MARGIN\_LEFT+30\*(RADIUS+1),MARGIN\_TOP,10,cxt);

renderDigit(MARGIN\_LEFT+39\*(RADIUS+1),MARGIN\_TOP,parseInt(minutes/10),cxt);

renderDigit(MARGIN\_LEFT+54\*(RADIUS+1),MARGIN\_TOP,parseInt(minutes%10),cxt);

renderDigit(MARGIN\_LEFT+69\*(RADIUS+1),MARGIN\_TOP,10,cxt);

renderDigit(MARGIN\_LEFT+78\*(RADIUS+1),MARGIN\_TOP,parseInt(seconds/10),cxt);

renderDigit(MARGIN\_LEFT+93\*(RADIUS+1),MARGIN\_TOP,parseInt(seconds%10),cxt);

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

cxt.fillStyle=balls[i].colors;

cxt.beginPath();

cxt.arc(balls[i].x,balls[i].y,RADIUS,0,2\*Math.PI,true);

cxt.closePath();

cxt.fill();

}

}

function renderDigit(x,y,num,cxt){

cxt.fillStyle = "rgb(0,102,153)";

for(var i=0;i<digit[num].length;i++){

for(var j=0;j<digit[num][i].length;j++){

if(digit[num][i][j]==1){

cxt.beginPath();

cxt.arc(x+j\*2\*(RADIUS+1)+(RADIUS+1),y+i\*2\*(RADIUS+1)+(RADIUS+1),RADIUS,0,2\*Math.PI);

cxt.closePath();

cxt.fill();

}

}

}

}

</script>

</body>

</html>

## 优化、扩展

### 性能优化

存在问题：

balls数组越来越长;

解决方法：

让滚出屏幕的小球从数组删除

### 屏幕自适应

记得把body和canvas的height设置为100%，使其撑开。

<!DOCTYPE html>

<html>

<head>

<title></title>

</head>

<body style="height:100%">

<canvas id="canvas" style="height:100%">

当前浏览器不支持Canvas

</canvas>

<script type="text/javascript">

var WINDOW\_WIDTH=1024;

var WINDOW\_HEIGHT=768;

var RADIUS =8;

var MARGIN\_TOP=60;

var MARGIN\_LEFT=30;

const endTime = new Date(2016,3,12,18,47,52);//设置截止日期2014-07

var curShowTimeSeconds = 0;

var balls = [];//生成小球

var colors = ["#33B5E5","#0099CC","#AA66CC","#9933CC","#99CC00","#669900","#FFBB33","#FF8800","#FF4444","#CC0000"];

digit =

[

[

[0,0,1,1,1,0,0],

[0,1,1,0,1,1,0],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,0,1,1,0],

[0,0,1,1,1,0,0]

],//0

[

[0,0,0,1,1,0,0],

[0,1,1,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[1,1,1,1,1,1,1]

],//1

[

[0,1,1,1,1,1,0],

[1,1,0,0,0,1,1],

[0,0,0,0,0,1,1],

[0,0,0,0,1,1,0],

[0,0,0,1,1,0,0],

[0,0,1,1,0,0,0],

[0,1,1,0,0,0,0],

[1,1,0,0,0,0,0],

[1,1,0,0,0,1,1],

[1,1,1,1,1,1,1]

],//2

[

[1,1,1,1,1,1,1],

[0,0,0,0,0,1,1],

[0,0,0,0,1,1,0],

[0,0,0,1,1,0,0],

[0,0,1,1,1,0,0],

[0,0,0,0,1,1,0],

[0,0,0,0,0,1,1],

[0,0,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,1,1,1,0]

],//3

[

[0,0,0,0,1,1,0],

[0,0,0,1,1,1,0],

[0,0,1,1,1,1,0],

[0,1,1,0,1,1,0],

[1,1,0,0,1,1,0],

[1,1,1,1,1,1,1],

[0,0,0,0,1,1,0],

[0,0,0,0,1,1,0],

[0,0,0,0,1,1,0],

[0,0,0,1,1,1,1]

],//4

[

[1,1,1,1,1,1,1],

[1,1,0,0,0,0,0],

[1,1,0,0,0,0,0],

[1,1,1,1,1,1,0],

[0,0,0,0,0,1,1],

[0,0,0,0,0,1,1],

[0,0,0,0,0,1,1],

[0,0,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,1,1,1,0]

],//5

[

[0,0,0,0,1,1,0],

[0,0,1,1,0,0,0],

[0,1,1,0,0,0,0],

[1,1,0,0,0,0,0],

[1,1,0,1,1,1,0],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,1,1,1,0]

],//6

[

[1,1,1,1,1,1,1],

[1,1,0,0,0,1,1],

[0,0,0,0,1,1,0],

[0,0,0,0,1,1,0],

[0,0,0,1,1,0,0],

[0,0,0,1,1,0,0],

[0,0,1,1,0,0,0],

[0,0,1,1,0,0,0],

[0,0,1,1,0,0,0],

[0,0,1,1,0,0,0]

],//7

[

[0,1,1,1,1,1,0],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,1,1,1,0],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,1,1,1,0]

],//8

[

[0,1,1,1,1,1,0],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[1,1,0,0,0,1,1],

[0,1,1,1,0,1,1],

[0,0,0,0,0,1,1],

[0,0,0,0,0,1,1],

[0,0,0,0,1,1,0],

[0,0,0,1,1,0,0],

[0,1,1,0,0,0,0]

],//9

[

[0,0,0,0],

[0,0,0,0],

[0,1,1,0],

[0,1,1,0],

[0,0,0,0],

[0,0,0,0],

[0,1,1,0],

[0,1,1,0],

[0,0,0,0],

[0,0,0,0]

]//:

];

window.onload = function(){

WINDOW\_WIDTH =document.documentElement.clientWidth||document.body.clientWidth;

WINDOW\_HEIGHT = document.documentElement.clientHeight||document.body.clientHeight;

MARGIN\_LEFT=Math.round(WINDOW\_WIDTH/10);

RADIUS =Math.round(WINDOW\_WIDTH\*4/5/108)-1;

MARGIN\_TOP=Math.round(WINDOW\_HEIGHT/5);

canvas = document.getElementById('canvas');

context = canvas.getContext('2d');

canvas.width = WINDOW\_WIDTH;

canvas.height = WINDOW\_HEIGHT;

curShowTimeSeconds = getCurrentShowTimeSeconds();

setInterval(function(){

render(context);

update();//对当前数据进行调整

},50)

}

function getCurrentShowTimeSeconds(){

var curTime = new Date();

var ret =endTime.getTime() - curTime.getTime();

ret = Math.round(ret/1000);

return ret>0?ret:0;

}

function update(){

var nextShowTimeSeconds = getCurrentShowTimeSeconds();

var nextHours = parseInt(nextShowTimeSeconds/3600);

var nextMinutes =parseInt((nextShowTimeSeconds-nextHours\*3600)/60);

var nextSeconds =parseInt(nextShowTimeSeconds%60);

var curHours =parseInt(curShowTimeSeconds/3600);

var curMinutes =parseInt((curShowTimeSeconds-curHours\*3600)/60);

var curSeconds =parseInt(curShowTimeSeconds%60);

if(nextSeconds!=curSeconds){

if(parseInt(curHours/10)!=parseInt(nextHours/10)){

addBalls(MARGIN\_LEFT+0,MARGIN\_TOP,parseInt(curHours/10));

}

if(parseInt(curHours%10)!=parseInt(nextHours%10)){

addBalls(MARGIN\_LEFT+15\*(RADIUS+1),MARGIN\_TOP,parseInt(curHours%10));

}

if(parseInt(curMinutes/10)!=parseInt(nextMinutes/10)){

addBalls(MARGIN\_LEFT+39\*(RADIUS+1),MARGIN\_TOP,parseInt(curMinutes/10));

}

if(parseInt(curMinutes%10)!=parseInt(nextMinutes%10)){

addBalls(MARGIN\_LEFT+54\*(RADIUS+1),MARGIN\_TOP,parseInt(curMinutes%10));

}

if(parseInt(curSeconds/10)!=parseInt(nextSeconds/10)){

addBalls(MARGIN\_LEFT+78\*(RADIUS+1),MARGIN\_TOP,parseInt(curSeconds/10));

}

if(parseInt(curSeconds%10)!=parseInt(nextSeconds%10)){

addBalls(MARGIN\_LEFT+93\*(RADIUS+1),MARGIN\_TOP,parseInt(curSeconds%10));

}

curShowTimeSeconds = nextShowTimeSeconds;

}

updateBalls();

}

function updateBalls () {

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

balls[i].x+=balls[i].vx;

balls[i].y+=balls[i].vy;

balls[i].vy+=balls[i].g;

if(balls[i].y>=WINDOW\_HEIGHT-RADIUS){

balls[i].y=WINDOW\_HEIGHT-RADIUS;

balls[i].vy = -balls[i].vy\*0.75;

}

}

var count = 0;

for(var i=0;i<balls.length;i++)

if(balls[i].x+RADIUS>0&&balls[i].x-RADIUS<WINDOW\_WIDTH)

balls[count++] = balls[i];

while(balls.length>count){

balls.pop();

}

}

function addBalls(x,y,num){

for(var i=0;i<digit[num].length;i++){

for(var j=0;j<digit[num][i].length;j++){

if(digit[num][i][j]==1){

var aBall={

x:x+j\*2\*(RADIUS+1)+(RADIUS+1),

y:y+i\*2\*(RADIUS+1)+(RADIUS+1),

g:1.5+Math.random(),

vx:Math.pow(-1,Math.ceil(Math.random()\*1000))\*4,

vy:-5,

colors:colors[Math.floor(Math.random()\*colors.length)]

}

balls.push(aBall);

}

}

}

}

function render(cxt){

cxt.clearRect(0,0,WINDOW\_WIDTH,WINDOW\_HEIGHT);//清空屏幕

var hours =parseInt(curShowTimeSeconds/3600);

var minutes =parseInt((curShowTimeSeconds-hours\*3600)/60);

var seconds =parseInt(curShowTimeSeconds%60);

renderDigit(MARGIN\_LEFT,MARGIN\_TOP,parseInt(hours/10),cxt);//一定要传绘图上下文环境

renderDigit(MARGIN\_LEFT+15\*(RADIUS+1),MARGIN\_TOP,parseInt(hours%10),cxt);

renderDigit(MARGIN\_LEFT+30\*(RADIUS+1),MARGIN\_TOP,10,cxt);

renderDigit(MARGIN\_LEFT+39\*(RADIUS+1),MARGIN\_TOP,parseInt(minutes/10),cxt);

renderDigit(MARGIN\_LEFT+54\*(RADIUS+1),MARGIN\_TOP,parseInt(minutes%10),cxt);

renderDigit(MARGIN\_LEFT+69\*(RADIUS+1),MARGIN\_TOP,10,cxt);

renderDigit(MARGIN\_LEFT+78\*(RADIUS+1),MARGIN\_TOP,parseInt(seconds/10),cxt);

renderDigit(MARGIN\_LEFT+93\*(RADIUS+1),MARGIN\_TOP,parseInt(seconds%10),cxt);

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

cxt.fillStyle=balls[i].colors;

cxt.beginPath();

cxt.arc(balls[i].x,balls[i].y,RADIUS,0,2\*Math.PI,true);

cxt.closePath();

cxt.fill();

}

}

function renderDigit(x,y,num,cxt){

cxt.fillStyle = "rgb(0,102,153)";

for(var i=0;i<digit[num].length;i++){

for(var j=0;j<digit[num][i].length;j++){

if(digit[num][i][j]==1){

cxt.beginPath();

cxt.arc(x+j\*2\*(RADIUS+1)+(RADIUS+1),y+i\*2\*(RADIUS+1)+(RADIUS+1),RADIUS,0,2\*Math.PI);

cxt.closePath();

cxt.fill();

}

}

}

}

</script>

</body>

</html>

### 改进成时间管理小工具

getTime()，获取距离1970-01-01 零点的毫秒数

var endTime = new Date();

endTime.setTime(endTime.getTime()+3600\*1000);//当前时间向后退1个小时

### 改进为绚丽的时钟

function getCurrentShowTimeSeconds(){

var curTime = new Date();

var ret = curTime.getHours()\*3600+curTime.getMinutes()\*60+curTime.getSeconds();

return ret;

//var ret =endTime.getTime() - curTime.getTime();

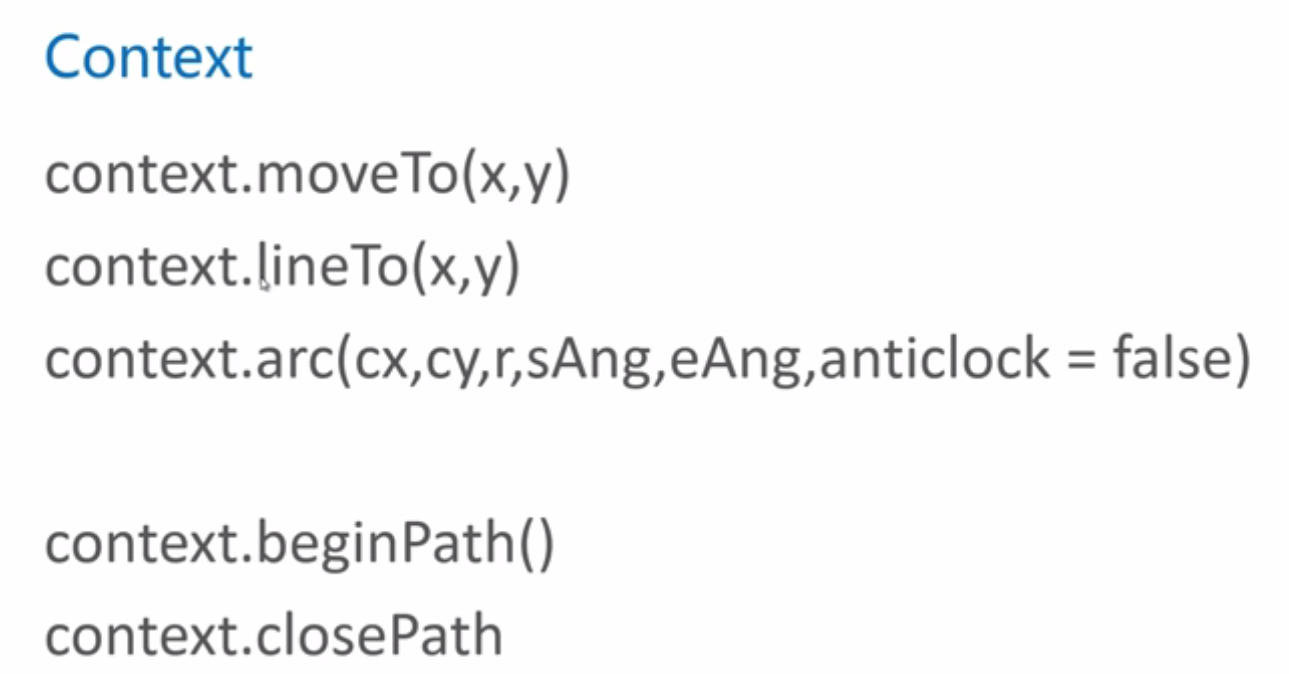
//ret = Math.round(ret/1000);

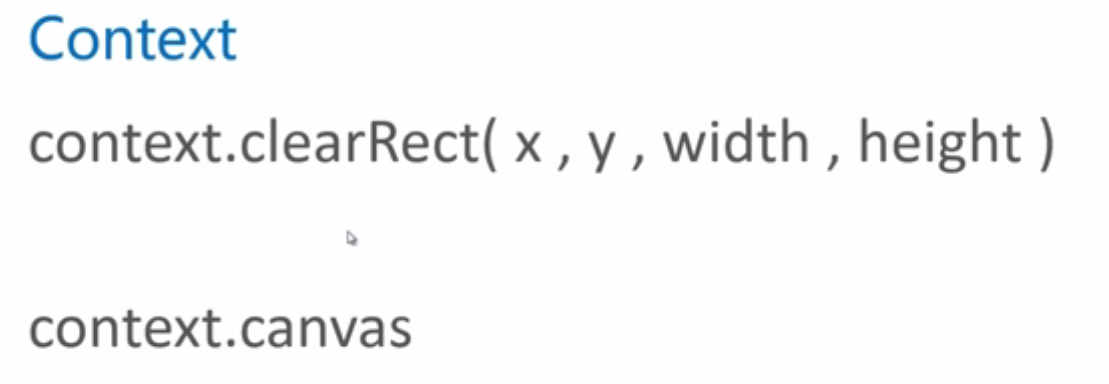
//return ret>0?ret:0;

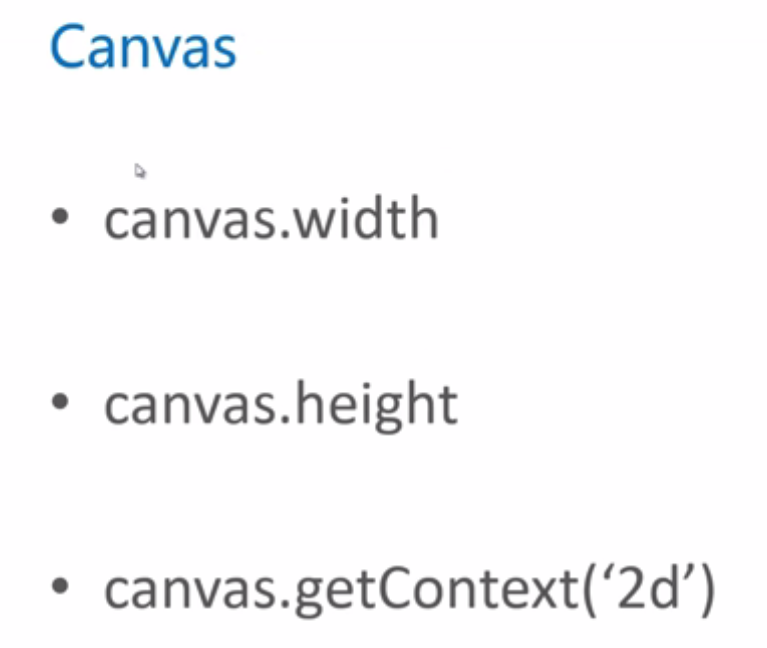
}

## 总结

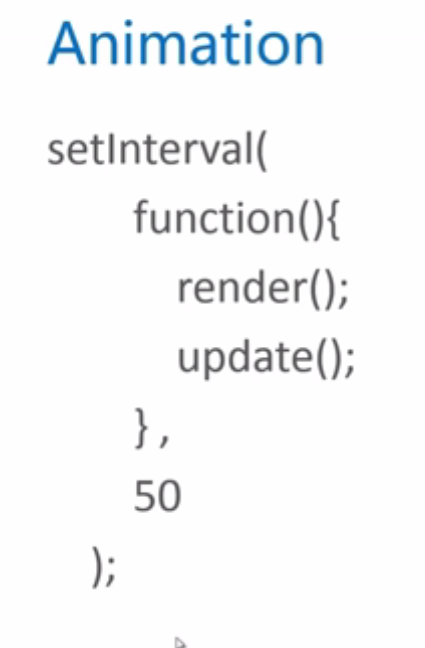








canvas绘制动画 一次渲染+一次底层数据更新

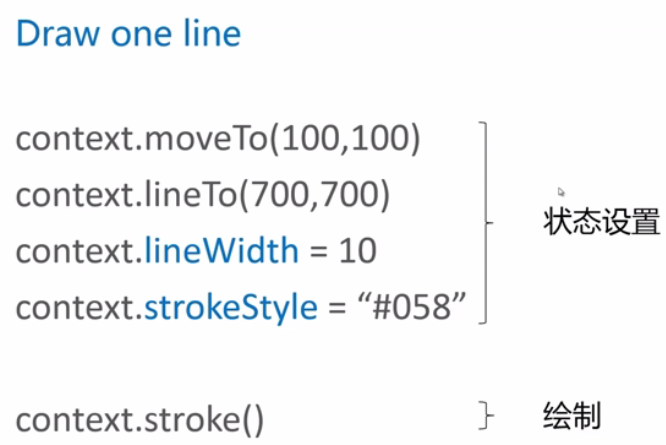


# class2Canvas绘图详解

## 从线条开始

### 回顾—从线条开始





lineWidth:线条宽度

strokeStyle：线条样式

是对上下文环境context进行绘制。

### 画一条直线

<!DOCTYPE html>

<html>

<head lang="en">

<meta charset="UTF-8">

<title></title>

</head>

<body>

<canvas id="canvas" style="border:1px solid #aaa;display:block;margin:50px auto;">

当前浏览器不支持Canvas，请更换浏览器后再试

</canvas>

<script>

window.onload = function(){

var canvas = document.getElementById("canvas");

canvas.width = 800;

canvas.height = 800;

context = canvas.getContext("2d");

context.moveTo(100,100);

context.lineTo(700,700);

context.lineWidth = 30;

context.strokeStyle="#058"

context.stroke();

}

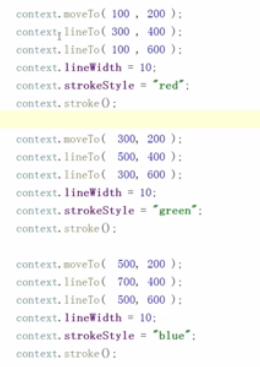
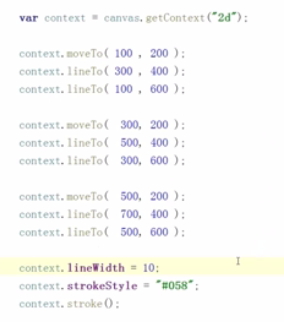
</script>

</body>

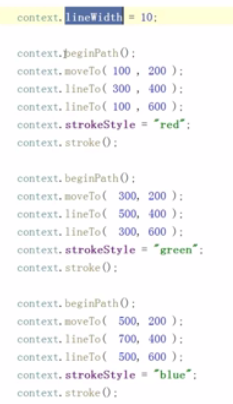
</html>

### 线条组成的图形和beginPath()

三条颜色相同的折线



三条颜色不同的折线



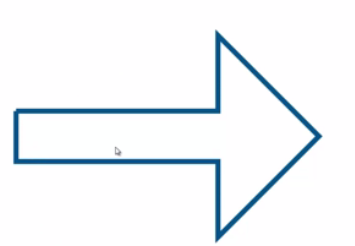
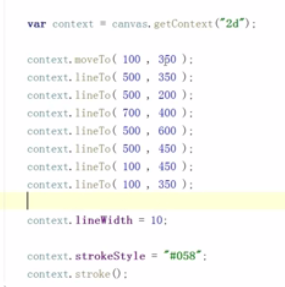
lineWidth一直使用，不会恢复到默认状态

beginPath()和LineTo在一起使用相当于moveTo.

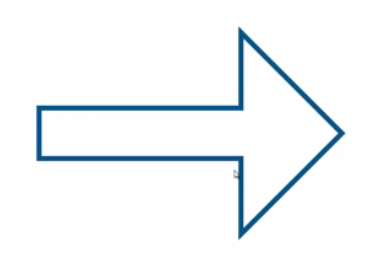
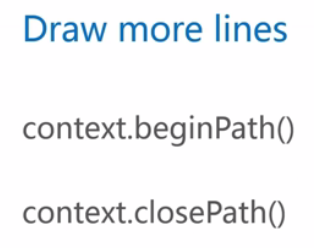
beginPath()：启动一个新的路径

### 多边形的填充和closePath

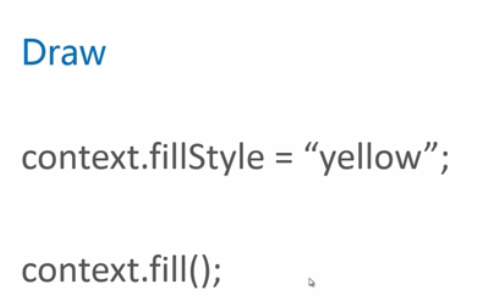
这样做的话，闭合的地方有瑕疵



绘制封闭多边形标准做法：



对封闭多边形进行填充【先填充，后描边】



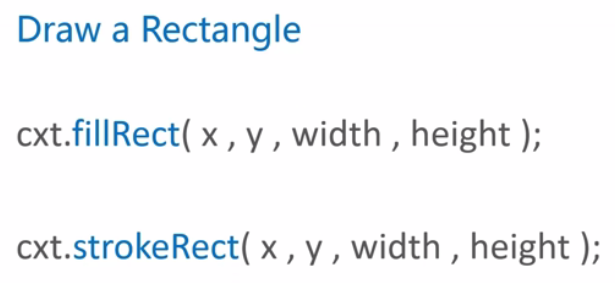
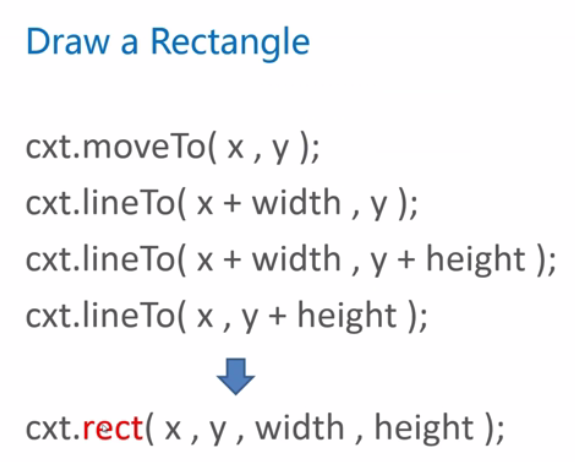
封装函数



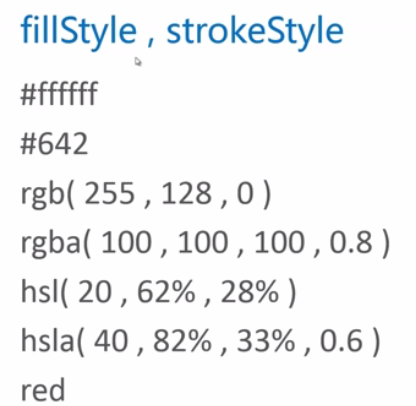
### 矩形、覆盖和透明色

fillRect和strokeRect使用的是fillStyle和stroke的样式

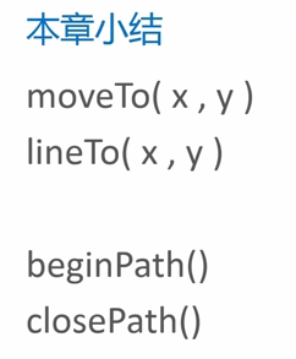
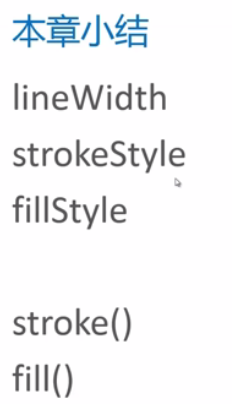
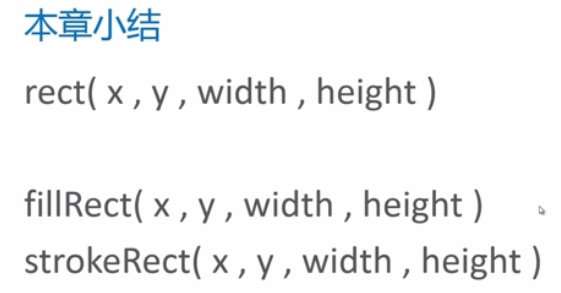
后绘制的图形如果和前面绘制的图形出现了重叠，后绘制的会遮挡前面绘制的图形。



fillStyle和strokeStyle可以使用的颜色系统：



### 小结：从线条开始

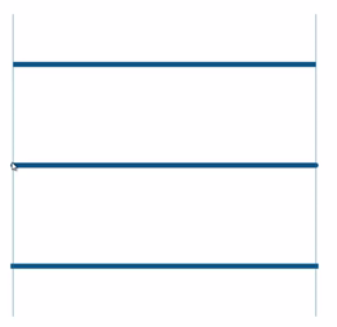
## 线条的属性

### 线条的帽子：lineCap

线条是组成绘图最基本的元素

线条两端的形状lineCap

只能用于线段的开始处和结束处，不能用于连接处。

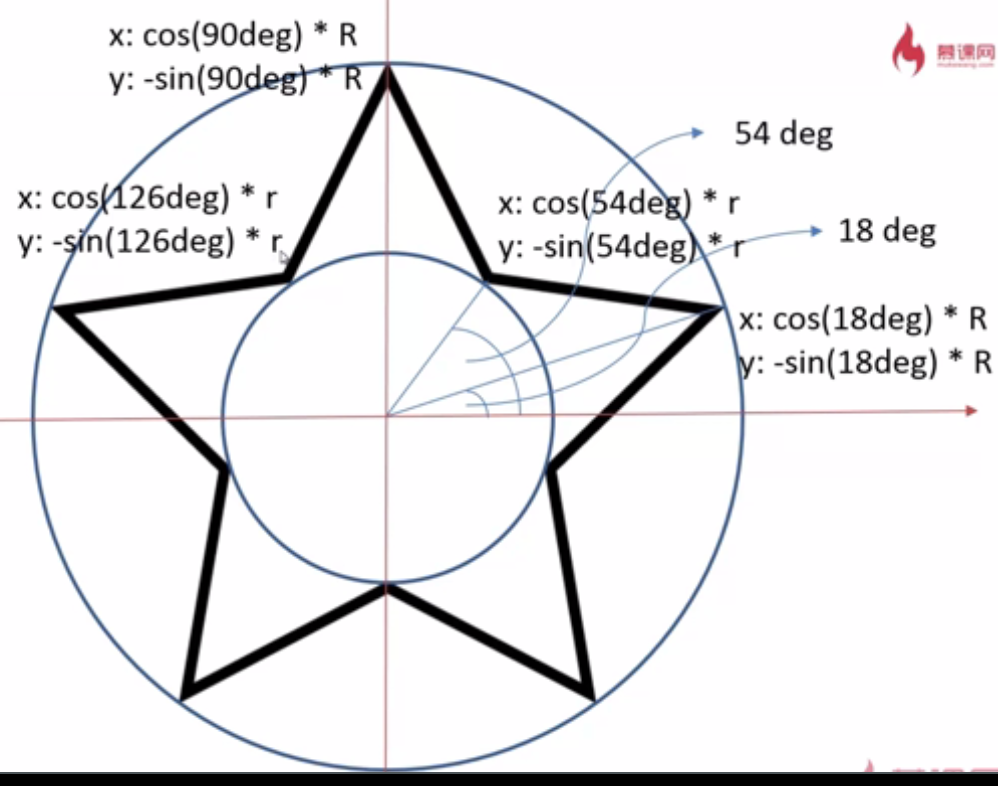
lineCap可以解决线条连接处的瑕疵问题【但还是建议用closePath来解决】



### 画一个五角星



思路分析



函数封装

function drawStart (cxt,r,R,x,y) {

cxt.beginPath();

for(var i=0;i<5;i++){

cxt.lineTo(Math.cos((18+i\*72)/180\*Math.PI)\*R+x,

-Math.sin((18+i\*72)/180\*Math.PI)\*R+y);

cxt.lineTo(Math.cos((54+i\*72)/180\*Math.PI)\*r+x,

-Math.sin((54+i\*72)/180\*Math.PI)\*r+y);

}

cxt.closePath();

cxt.stroke();

}

加入旋转

function drawStart (cxt,r,R,x,y,rot) {

cxt.beginPath();

for(var i=0;i<5;i++){

cxt.lineTo(Math.cos((18+i\*72-rot)/180\*Math.PI)\*R+x,

-Math.sin((18+i\*72-rot)/180\*Math.PI)\*R+y);

cxt.lineTo(Math.cos((54+i\*72-rot)/180\*Math.PI)\*r+x,

-Math.sin((54+i\*72-rot)/180\*Math.PI)\*r+y);

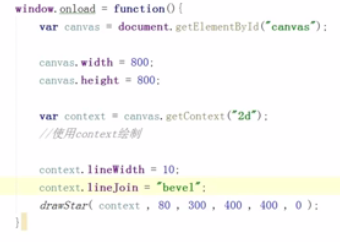
}

cxt.closePath();

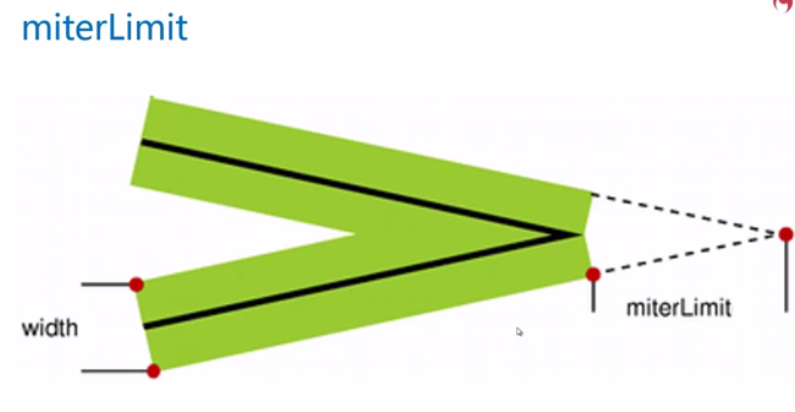
cxt.stroke();

}

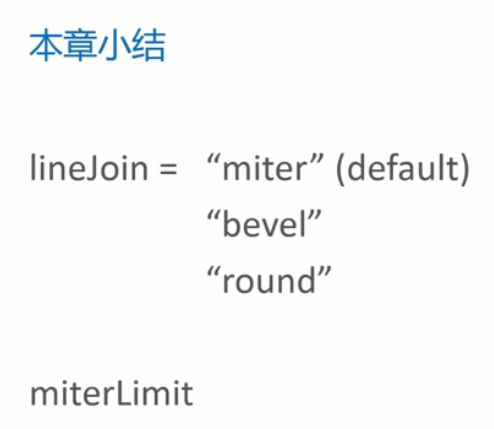
### 线条的连接：linejoin和miterLimit



**miterLimit只有在lineJoin为miter时才有作用**



### 小结



超过miterLimit之后，会自动转换为bevel形式

## 星空和图形变换

### 画一片星空

设置随机的半径，旋转角度等，用Math.random()

待优化：有些星星重叠，有些星星的一部分在外面

window.onload = function(){

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

canvas.width = 800;

canvas.height = 800;

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

context.fillStyle = "black";

context.fillRect(0,0,canvas.width,canvas.height);

for(var i=0;i<200;i++){

var r = Math.random()\*10+10;

var x = Math.random()\*canvas.width;

var y = Math.random()\*canvas.height;

var a = Math.random()\*360;

drawStart(context,x,y,r,r/2.0,a);

}

}

function drawStart (cxt,x,y,R,r,rot) {

cxt.beginPath();

for(var i=0;i<5;i++){

cxt.lineTo(Math.cos((18+i\*72-rot)/180\*Math.PI)\*R+x,

-Math.sin((18+i\*72-rot)/180\*Math.PI)\*R+y);

cxt.lineTo(Math.cos((54+i\*72-rot)/180\*Math.PI)\*r+x,

-Math.sin((54+i\*72-rot)/180\*Math.PI)\*r+y);

}

cxt.closePath();

cxt.fillStyle = "#fh3";

cxt.strokeStyle = "#fd5";

cxt.lineWidth = 3;

cxt.lineJoin = "round";

cxt.fill();

cxt.stroke();

}

### 图像变化和状态保存

思考：

突然想画一个除五角星的形状~

那已经封装的函数需要修改。

思路：

先绘制基本轮廓，再应用图形变换



window.onload = function(){

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

canvas.width = 800;

canvas.height = 800;

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

context.fillStyle = "red";

context.translate(100,100);

context.fillRect(0,0,400,400);

}

陷阱：图形变换函数是叠加的

window.onload = function(){

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

canvas.width = 800;

canvas.height = 800;

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

context.fillStyle = "red";

context.translate(100,100);

context.fillRect(0,0,400,400);

context.fillStyle = "green";

context.translate(300,300);

context.fillRect(0,0,400,400);

}

解决方法1：进行反变换

window.onload = function(){

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

canvas.width = 800;

canvas.height = 800;

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

context.fillStyle = "red";

context.translate(100,100);

context.fillRect(0,0,400,400);

context.translate(-100,-100);

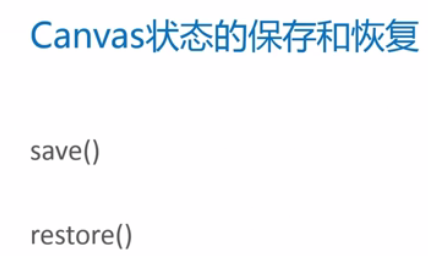
context.fillStyle = "green";

context.translate(300,300);

context.fillRect(0,0,400,400);

}

解决方法2：context.save(),context.restore();



window.onload = function(){

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

canvas.width = 800;

canvas.height = 800;

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

context.save();

context.fillStyle = "red";

context.translate(100,100);

context.fillRect(0,0,400,400);

context.restore();

context.save();

context.fillStyle = "green";

context.translate(300,300);

context.fillRect(0,0,400,400);

context.restore();

}

### 应用translate、rotate和scale

window.onload = function(){

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

canvas.width = 800;

canvas.height = 800;

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

context.fillStyle = "black";

context.fillRect(0,0,canvas.width,canvas.height);

for(var i=0;i<200;i++){

var r = Math.random()\*10+10;

var x = Math.random()\*canvas.width;

var y = Math.random()\*canvas.height;

var a = Math.random()\*360;

drawStart(context,x,y,r,a);

}

}

function drawStart (cxt,x,y,R,rot) {

cxt.save();

cxt.translate(x,y);

cxt.rotate(rot/180\*Math.PI);

startPath(cxt);//标准的五角星

cxt.fillStyle = "#fb3";

cxt.strokeStyle = "#fd5";

cxt.lineWidth = 3;

cxt.lineJoin = "round"

cxt.fill();

cxt.stroke();

cxt.restore();

}

function startPath(cxt){

cxt.beginPath();

for(var i=0;i<5;i++){

cxt.lineTo(Math.cos((18+i\*72)/180\*Math.PI)\*20,

-Math.sin((18+i\*72)/180\*Math.PI)\*20);

cxt.lineTo(Math.cos((54+i\*72)/180\*Math.PI)\*0.5\*20,

-Math.sin((54+i\*72)/180\*Math.PI)\*0.5\*20);

}

cxt.closePath();

}

scale的陷阱：不仅放大图像大小，对其它数值属性也会进行放大，比如起点，边框

window.onload = function(){

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

canvas.width = 800;

canvas.height = 800;

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

context.fillStyle = "black";

context.fillRect(0,0,canvas.width,canvas.height);

for(var i=0;i<200;i++){

var r = Math.random()\*10+10;

var x = Math.random()\*canvas.width;

var y = Math.random()\*canvas.height;

var a = Math.random()\*360;

drawStart(context,x,y,r,a);

}

}

function drawStart (cxt,x,y,R,rot) {

cxt.save();

cxt.translate(x,y);

cxt.rotate(rot/180\*Math.PI);

cxt.scale(R,R);

startPath(cxt);//标准的五角星

cxt.fillStyle = "#fb3";

//cxt.strokeStyle = "#fd5";

//cxt.lineWidth = 3;

//cxt.lineJoin = "round"

cxt.fill();

//cxt.stroke();

cxt.restore();

}

function startPath(cxt){

cxt.beginPath();

for(var i=0;i<5;i++){

cxt.lineTo(Math.cos((18+i\*72)/180\*Math.PI),

-Math.sin((18+i\*72)/180\*Math.PI));

cxt.lineTo(Math.cos((54+i\*72)/180\*Math.PI)\*0.5,

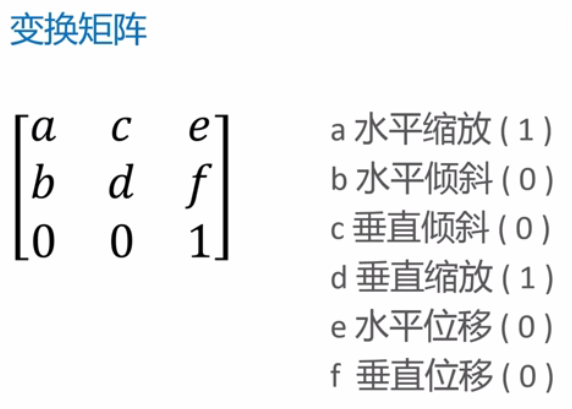
-Math.sin((54+i\*72)/180\*Math.PI)\*0.5);

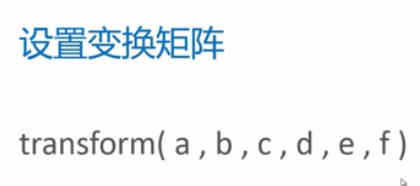
}

cxt.closePath();

}

### 深入理解图形变换

，不进行变化时，变化矩阵是一个单位矩阵



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

canvas.width = 1200;

canvas.height = 800;

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

context.fillStyle = "red";

context.strokeStyle = "#058";

context.lineWidth = 5;

context.save();

context.transform(2,0.2,0.2,1.5,50,100);

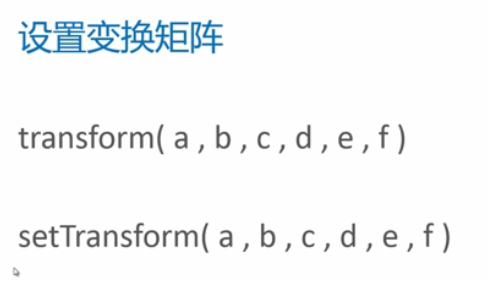
context.fillRect(50,50,300,300);

context.strokeRect(50,50,300,300);

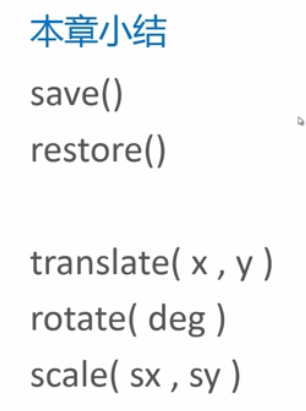
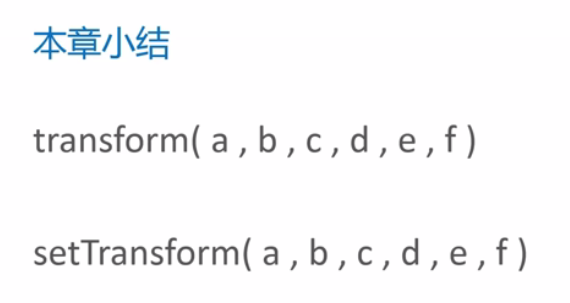
context.restore();

transform()会产生级联的效果

setTransform()会使前面设置的transform()失效；

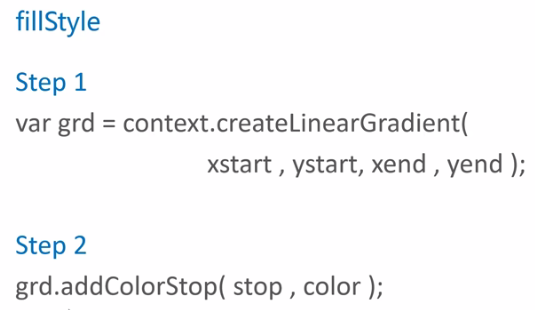


### 小结：星空和图形变换

## 填充样式

### 线性渐变



例子：

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

canvas.width = 1200;

canvas.height = 800;

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

var linearGrad = context.createLinearGradient(0,0,800,800);//渐变色是倾斜的

//var linearGrad = context.createLinearGradient(0,0,800,0);//渐变色是水平的

linearGrad.addColorStop(0.0,"#fff");

linearGrad.addColorStop(1.0,"#000");

context.fillStyle = linearGrad;

context.fillRect(0,0,800,800);

如果渐变区域设置得比绘制矩形小，那么剩下的部分用最后一个stopcolor填充

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

canvas.width = 800;

canvas.height = 800;

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

var linearGrad = context.createLinearGradient(0,0,400,400);

linearGrad.addColorStop(0.0,"white");

linearGrad.addColorStop(0.25,"yellow");

linearGrad.addColorStop(0.5,"green");

linearGrad.addColorStop(0.75,"blue");

linearGrad.addColorStop(1.0,"black");

context.fillStyle = linearGrad;

context.fillRect(0,0,800,800);

如果渐变区域设置得比绘制矩形大，那么有些色就消失。

设置星空的背景

window.onload = function(){

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

canvas.width = 1200;

canvas.height = 800;

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

var skyStyle = context.createLinearGradient(0,0,0,canvas.height);

skyStyle.addColorStop(0.0,'black');

skyStyle.addColorStop(1.0,'#035');

context.fillStyle = skyStyle;

context.fillRect(0,0,canvas.width,canvas.height);

for(var i=0;i<200;i++){

var r = Math.random()\*5+5;

var x = Math.random()\*canvas.width;

var y = Math.random()\*canvas.height\*0.65;

var a = Math.random()\*360;

drawStart(context,x,y,r,a);

}

}

function drawStart (cxt,x,y,R,rot) {

cxt.save();

cxt.translate(x,y);

cxt.rotate(rot/180\*Math.PI);

cxt.scale(R,R);

startPath(cxt);//标准的五角星

cxt.fillStyle = "#fb3";

//cxt.strokeStyle = "#fd5";

//cxt.lineWidth = 3;

//cxt.lineJoin = "round"

cxt.fill();

//cxt.stroke();

cxt.restore();

}

function startPath(cxt){

cxt.beginPath();

for(var i=0;i<5;i++){

cxt.lineTo(Math.cos((18+i\*72)/180\*Math.PI),

-Math.sin((18+i\*72)/180\*Math.PI));

cxt.lineTo(Math.cos((54+i\*72)/180\*Math.PI)\*0.5,

-Math.sin((54+i\*72)/180\*Math.PI)\*0.5);

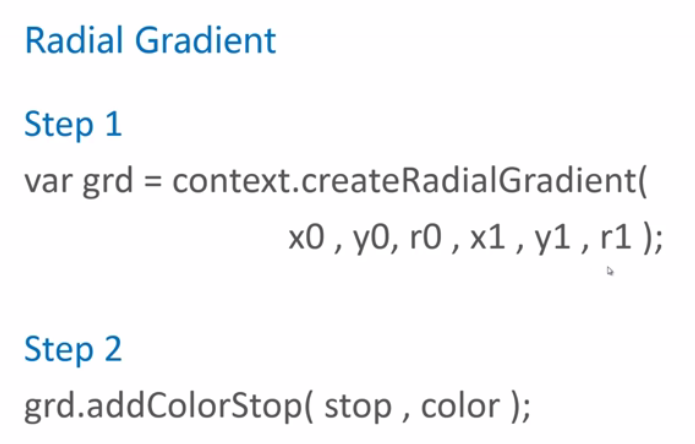
}

cxt.closePath();

}

### 径向渐变

定义在两个同心圆之间



<script type="text/javascript">

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

canvas.width = 800;

canvas.height = 800;

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

var linearGrad = context.createRadialGradient(400,400,100,400,400,500);

linearGrad.addColorStop(0.0,"white");

linearGrad.addColorStop(0.25,"yellow");

linearGrad.addColorStop(0.5,"green");

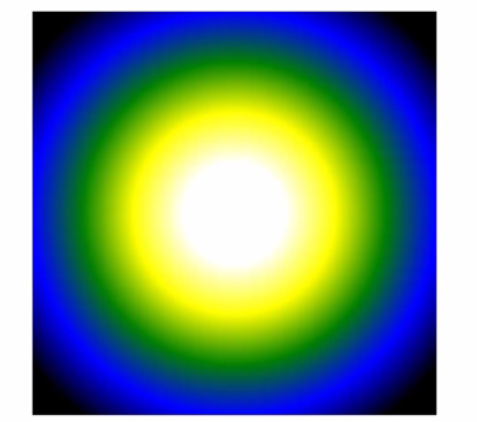
linearGrad.addColorStop(0.75,"blue");

linearGrad.addColorStop(1.0,"black");

context.fillStyle = linearGrad;

context.fillRect(0,0,800,800);

</script>



将星空改为径向渐变

<script type="text/javascript">

window.onload = function(){

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

canvas.width = 1200;

canvas.height = 800;

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

var skyStyle = context.createRadialGradient(canvas.width/2,canvas.height,0,canvas.width/2,canvas.height,canvas.height);

skyStyle.addColorStop(0.0,'#035');

skyStyle.addColorStop(1.0,'black');

context.fillStyle = skyStyle;

context.fillRect(0,0,canvas.width,canvas.height);

for(var i=0;i<200;i++){

var r = Math.random()\*5+5;

var x = Math.random()\*canvas.width;

var y = Math.random()\*canvas.height\*0.65;

var a = Math.random()\*360;

drawStart(context,x,y,r,a);

}

}

function drawStart (cxt,x,y,R,rot) {

cxt.save();

cxt.translate(x,y);

cxt.rotate(rot/180\*Math.PI);

cxt.scale(R,R);

startPath(cxt);//标准的五角星

cxt.fillStyle = "#fb3";

//cxt.strokeStyle = "#fd5";

//cxt.lineWidth = 3;

//cxt.lineJoin = "round"

cxt.fill();

//cxt.stroke();

cxt.restore();

}

function startPath(cxt){

cxt.beginPath();

for(var i=0;i<5;i++){

cxt.lineTo(Math.cos((18+i\*72)/180\*Math.PI),

-Math.sin((18+i\*72)/180\*Math.PI));

cxt.lineTo(Math.cos((54+i\*72)/180\*Math.PI)\*0.5,

-Math.sin((54+i\*72)/180\*Math.PI)\*0.5);

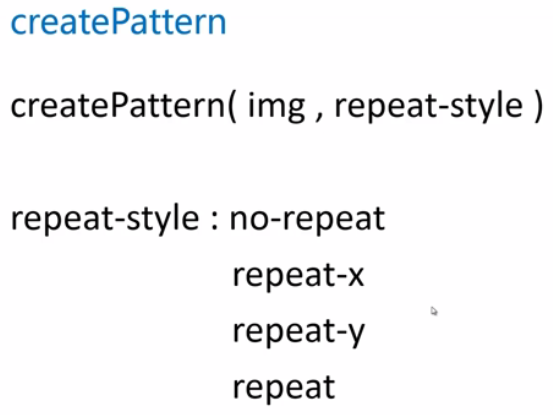
}

cxt.closePath();

}

</script>

### 使用图片、画布或者video



<script type="text/javascript">

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

canvas.width = 800;

canvas.height = 800;

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

var backgroundImage = new Image();

backgroundImage.src = "brick-s.jpg";

backgroundImage.onload = function(){

var pattern = context.createPattern(backgroundImage,'repeat');

context.fillStyle = pattern;

context.fillRect(0,0,800,800);

}

</script>

除了用图片进行填充，还可以用画布进行填充



<script type="text/javascript">

window.onload = function(){

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

canvas.width = 800;

canvas.height = 800;

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

var backCanvas= createBackgroundCanvas();

var pattern = context.createPattern(backCanvas,'repeat');

context.fillStyle = pattern;

context.fillRect(0,0,800,800);

}

function createBackgroundCanvas () {

var backCanvas = document.createElement('canvas');

backCanvas.width = 100;

backCanvas.height = 100;

var backCanvasContext = backCanvas.getContext('2d');

drawStar(backCanvasContext,50,50,50,0);

return backCanvas;

// body...

}

function drawStar (cxt,x,y,R,rot) {

cxt.save();

cxt.translate(x,y);

cxt.rotate(rot/180\*Math.PI);

cxt.scale(R,R);

startPath(cxt);//标准的五角星

cxt.fillStyle = "#fb3";

//cxt.strokeStyle = "#fd5";

//cxt.lineWidth = 3;

//cxt.lineJoin = "round"

cxt.fill();

//cxt.stroke();

cxt.restore();

}

function startPath(cxt){

cxt.beginPath();

for(var i=0;i<5;i++){

cxt.lineTo(Math.cos((18+i\*72)/180\*Math.PI),

-Math.sin((18+i\*72)/180\*Math.PI));

cxt.lineTo(Math.cos((54+i\*72)/180\*Math.PI)\*0.5,

-Math.sin((54+i\*72)/180\*Math.PI)\*0.5);

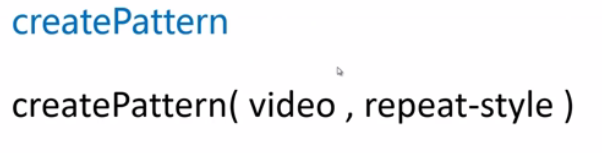
}

cxt.closePath();

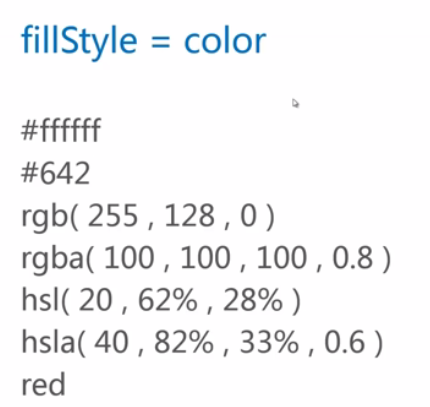
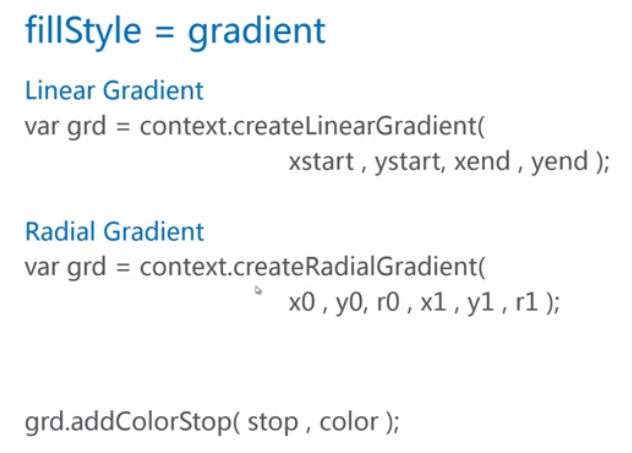
}

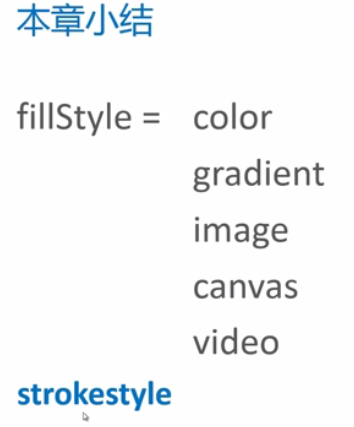
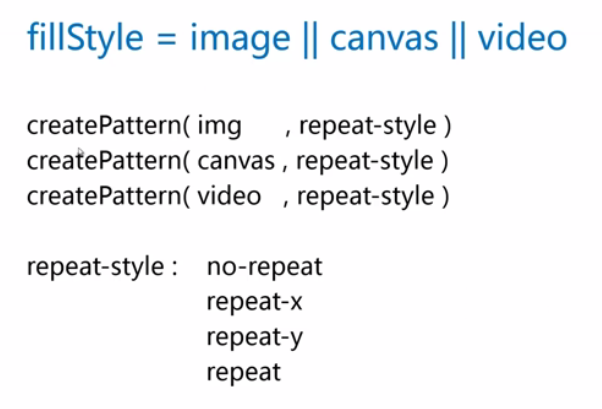
</script>

除了用图片进行填充，还可以用video进行填充



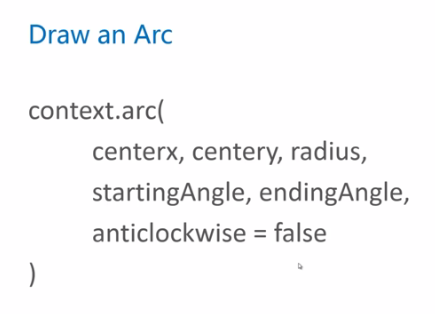
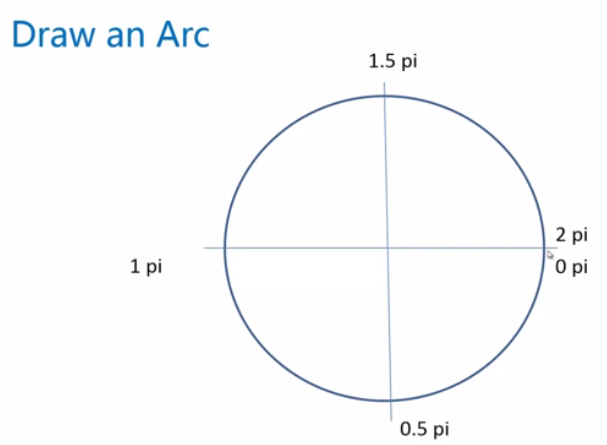
### 渐变色和纹理小结

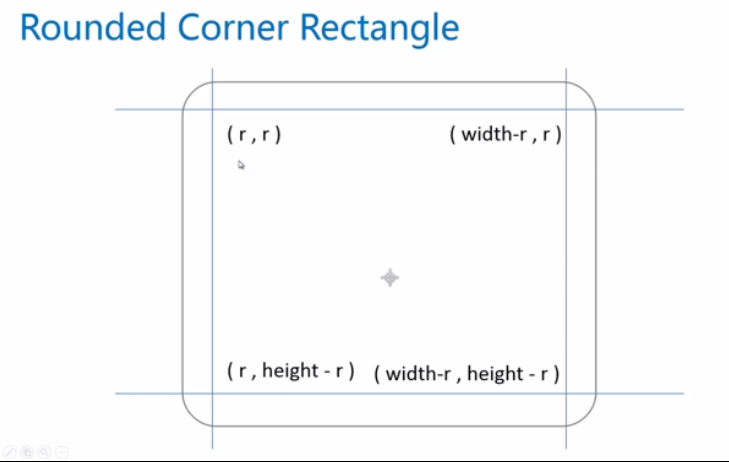


## 曲线的绘制

### arc()，圆弧和圆角矩形

例子：绘制圆角矩形



<script type="text/javascript">

window.onload = function(){

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

canvas.width = 800;

canvas.height = 800;

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

drawRoundRect(context,100,100,600,500,50)

}

function drawRoundRect(cxt,x,y,width,height,radius){

cxt.save();

cxt.translate(x,y);

pathRoundRect(cxt,width,height,radius);

cxt.strokeStyle = "black";

cxt.stroke();

cxt.restore();

}

function pathRoundRect (cxt,width,height,radius) {

cxt.beginPath();

cxt.arc(width-radius,height-radius,radius,0,Math.PI/2);

cxt.lineTo(radius,height);

cxt.arc(radius,height-radius,radius,Math.PI/2,Math.PI);

cxt.lineTo(0,radius);

cxt.arc(radius,radius,radius,Math.PI,Math.PI\*3/2);

cxt.lineTo(width-radius,0);

cxt.arc(width-radius,radius,radius,Math.PI\*3/2,Math.PI\*2);

cxt.closePath();

// body...

}

</script>

2048棋盘

<script type="text/javascript">

window.onload = function(){

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

canvas.width = 800;

canvas.height = 800;

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

fillRoundRect(context,150,150,500,500,10,'#bbada0');

for(var i=0;i<4;i++){

for(var j=0;j<4;j++){

fillRoundRect(context,170+i\*120,170+j\*120,100,100,6,"#ccc0b3");

}

}

}

function fillRoundRect(cxt,x,y,width,height,radius,fillColor){

if(2\*radius>width||2\*radius>height)

return;

cxt.save();

cxt.translate(x,y);

pathRoundRect(cxt,width,height,radius);

cxt.fillStyle = fillColor||"black";

cxt.fill();

cxt.restore();

}

function strokeRoundRect(cxt,x,y,width,height,radius,lineWidth,strokeColor){

if(2\*radius>width||2\*radius>height)

return;

cxt.save();

cxt.translate(x,y);

pathRoundRect(cxt,width,height,radius);

cxt.lineWidth = lineWidth||1;

cxt.strokeStyle = strokeColor||"black"

cxt.stroke();

cxt.restore();

}

function pathRoundRect (cxt,width,height,radius) {

cxt.beginPath();

cxt.arc(width-radius,height-radius,radius,0,Math.PI/2);

cxt.lineTo(radius,height);

cxt.arc(radius,height-radius,radius,Math.PI/2,Math.PI);

cxt.lineTo(0,radius);

cxt.arc(radius,radius,radius,Math.PI,Math.PI\*3/2);

cxt.lineTo(width-radius,0);

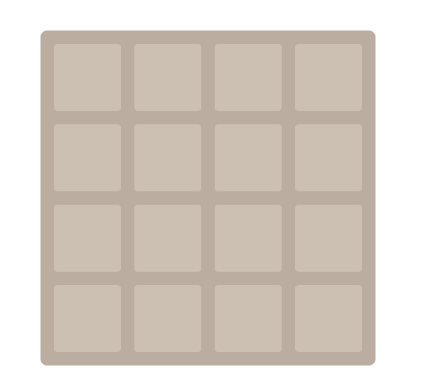
cxt.arc(width-radius,radius,radius,Math.PI\*3/2,Math.PI\*2);

cxt.closePath();

// body...

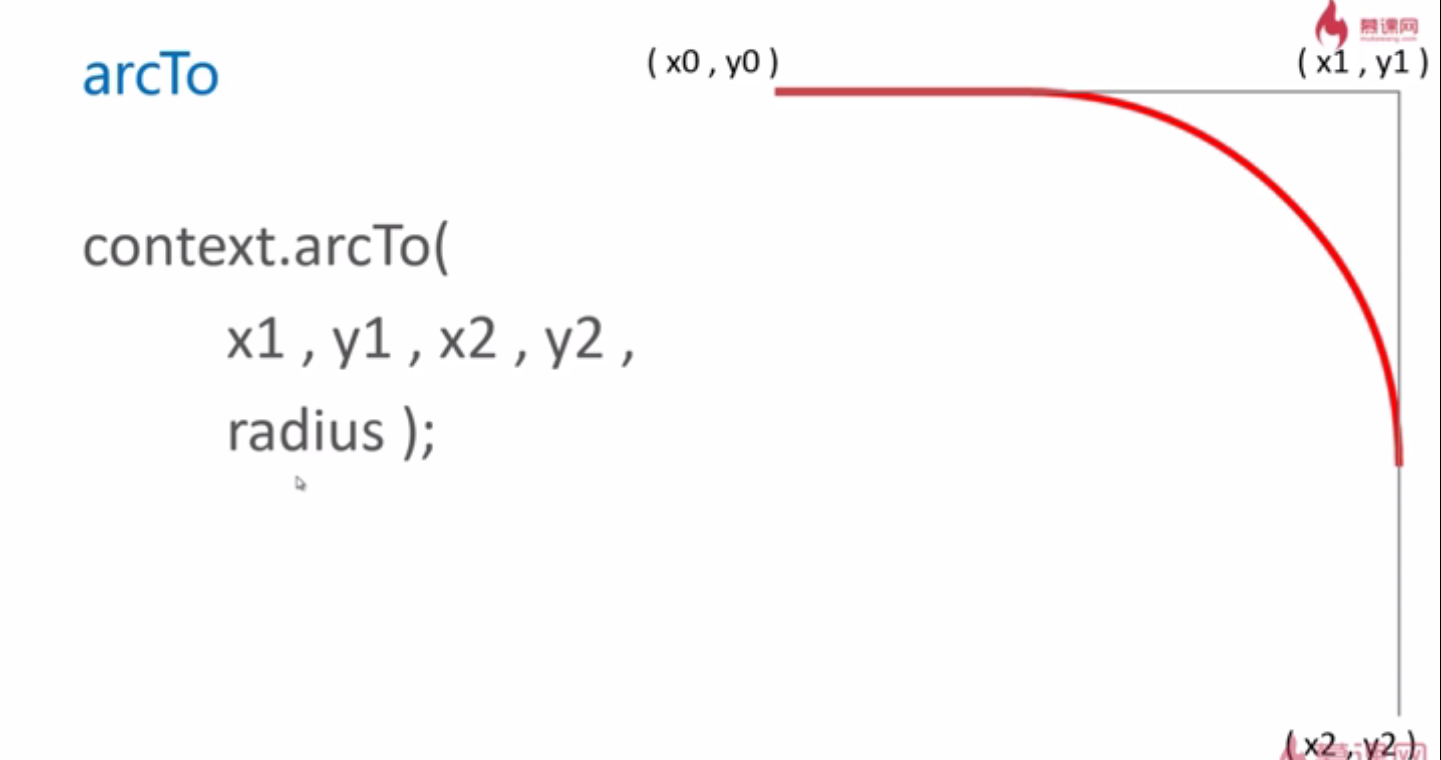
}

</script>



### arcTo()，另一种弧线绘制方式

控制点（x0,y0）



<script type="text/javascript">

window.onload = function(){

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

canvas.width = 800;

canvas.height = 800;

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

context.beginPath();

context.moveTo(150,150);

context.arcTo(650,150,650,640,300);

context.lineWidth = 6;

context.strokeStyle = "red";

context.stroke();

context.beginPath();

context.moveTo(150,150);

context.lineTo(650,150);

context.lineTo(650,650);

context.lineWidth = 2;

context.strokeStyle = "gray";

context.stroke();

}

</script>

对函数进行封装

<script type="text/javascript">

window.onload = function(){

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

canvas.width = 800;

canvas.height = 800;

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

arcToTest(context,150,100,650,100,650,600,100);

}

function arcToTest(cxt,x0,y0,x1,y1,x2,y2,R){

cxt.beginPath();

cxt.moveTo(x0,y0);

cxt.arcTo(x1,y1,x2,y2,R);

cxt.lineWidth = 6;

cxt.strokeStyle = "red";

cxt.stroke();

cxt.beginPath();

cxt.moveTo(x0,y0);

cxt.lineTo(x1,y1);

cxt.lineTo(x2,y2);

cxt.lineWidth = 2;

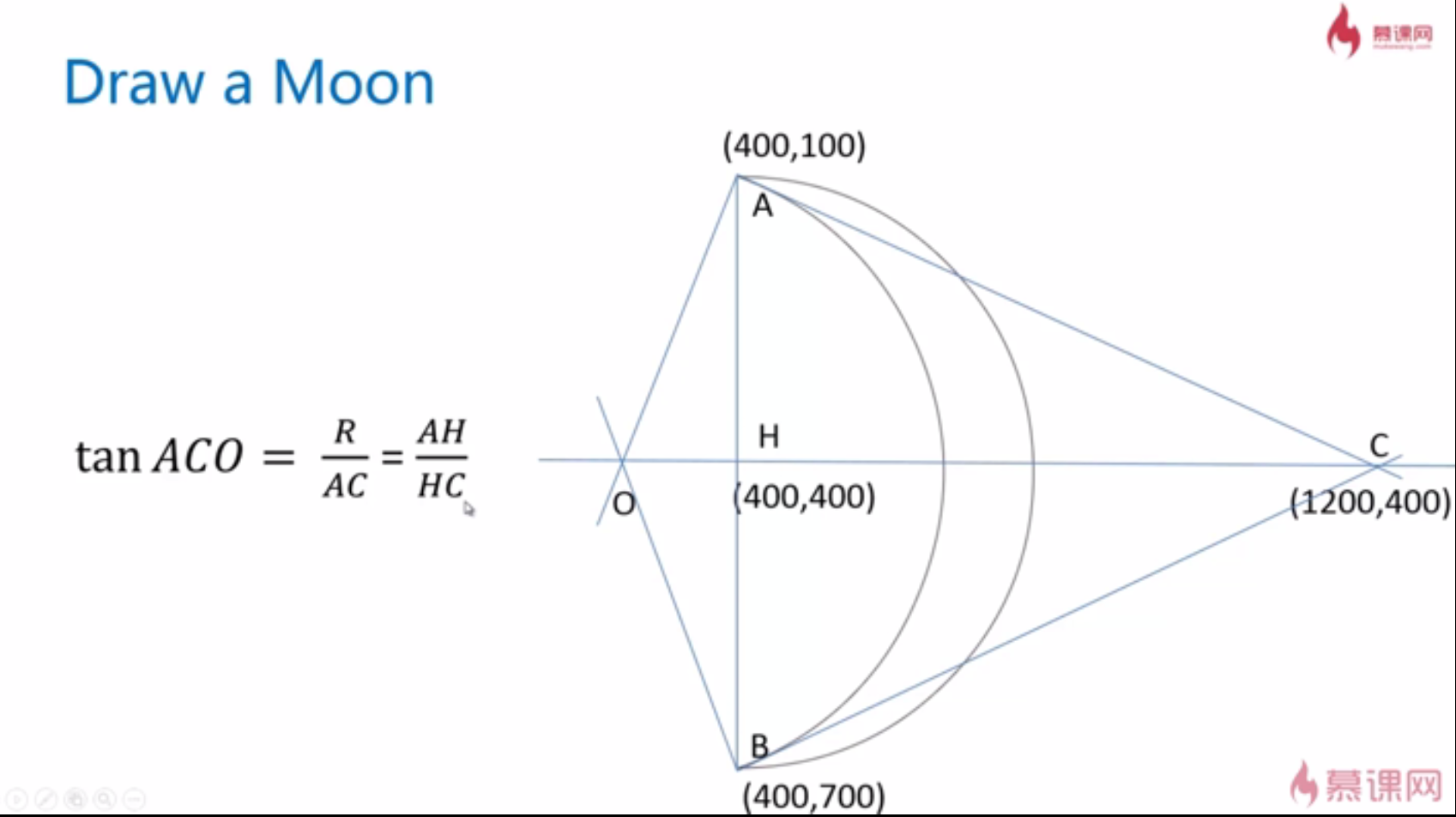
cxt.strokeStyle = "gray";

cxt.stroke();

}

</script>

### 绘制一角弯月



<!DOCTYPE html>

<html>

<head>

<title></title>

</head>

<body style="height:100%">

<canvas id="canvas" style="height:100%">

当前浏览器不支持Canvas

</canvas>

<script type="text/javascript">

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

canvas.width = 800;

canvas.height = 800;

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

var linearGrad = context.createLinearGradient(0,0,400,400);

linearGrad.addColorStop(0.0,"white");

linearGrad.addColorStop(0.25,"yellow");

linearGrad.addColorStop(0.5,"green");

linearGrad.addColorStop(0.75,"blue");

linearGrad.addColorStop(1.0,"black");

context.fillStyle = linearGrad;

context.fillRect(0,0,800,800);

window.onload = function(){

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

canvas.width = 1200;

canvas.height = 800;

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

var skyStyle = context.createLinearGradient(0,0,0,canvas.height);

skyStyle.addColorStop(0.0,'black');

skyStyle.addColorStop(1.0,'#035');

context.fillStyle = skyStyle;

context.fillRect(0,0,canvas.width,canvas.height);

for(var i=0;i<200;i++){

var r = Math.random()\*5+5;

var x = Math.random()\*canvas.width;

var y = Math.random()\*canvas.height\*0.65;

var a = Math.random()\*360;

drawStart(context,x,y,r,a);

}

fillMoon(context,2,900,200,100,30);

}

function drawStart (cxt,x,y,R,rot) {

cxt.save();

cxt.translate(x,y);

cxt.rotate(rot/180\*Math.PI);

cxt.scale(R,R);

startPath(cxt);//标准的五角星

cxt.fillStyle = "#fb3";

//cxt.strokeStyle = "#fd5";

//cxt.lineWidth = 3;

//cxt.lineJoin = "round"

cxt.fill();

//cxt.stroke();

cxt.restore();

}

function startPath(cxt){

cxt.beginPath();

for(var i=0;i<5;i++){

cxt.lineTo(Math.cos((18+i\*72)/180\*Math.PI),

-Math.sin((18+i\*72)/180\*Math.PI));

cxt.lineTo(Math.cos((54+i\*72)/180\*Math.PI)\*0.5,

-Math.sin((54+i\*72)/180\*Math.PI)\*0.5);

}

cxt.closePath();

}

function fillMoon(cxt,d,x,y,R,rot,fillColor){

cxt.save();

cxt.translate(x,y);

cxt.rotate(rot\*Math.PI/180);

cxt.scale(R,R);

pathMoon(cxt,d);

cxt.fillStyle = fillColor||"#fb5";

cxt.fill();

cxt.restore();

}

function pathMoon(cxt,d){

cxt.beginPath();

cxt.arc(0,0,1,0.5\*Math.PI,1.5\*Math.PI,true);

cxt.moveTo(0,-1);

cxt.arcTo(d,0,0,1,dist(0,-1,d,0)/d);

cxt.closePath();

}

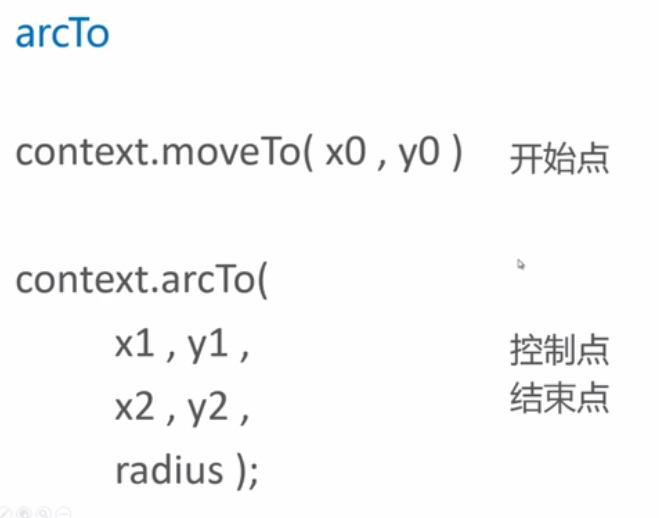
function dist(x1,y1,x2,y2){

return Math.sqrt((x1-x2)\*(x1-x2)+(y1-y2)\*(y1-y2));

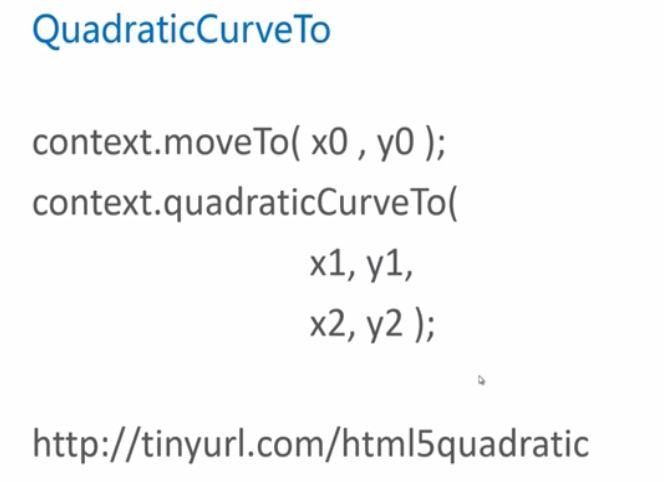
}

</script>

### quadraticCurveTo()，二次贝塞尔曲线



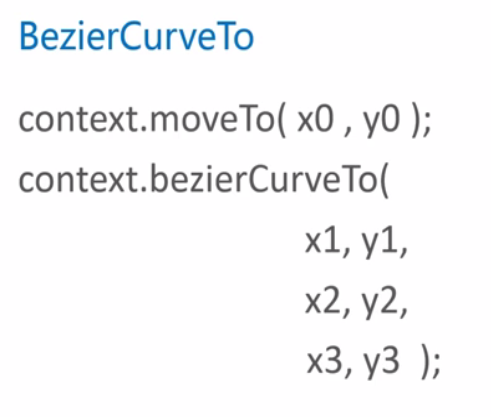
quadraticCurve开始点和结束点就是曲线的开始点和结束点



### bezierCurveTo()，三次贝塞尔曲线

二次贝塞尔画不出波浪。

唯一的区别，三次贝塞尔曲线拥有2个控制点



给星空加上绿地

<!DOCTYPE html>

<html>

<head>

<title></title>

</head>

<body style="height:100%">

<canvas id="canvas" style="height:100%">

当前浏览器不支持Canvas

</canvas>

<script type="text/javascript">

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

canvas.width = 800;

canvas.height = 800;

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

var linearGrad = context.createLinearGradient(0,0,400,400);

linearGrad.addColorStop(0.0,"white");

linearGrad.addColorStop(0.25,"yellow");

linearGrad.addColorStop(0.5,"green");

linearGrad.addColorStop(0.75,"blue");

linearGrad.addColorStop(1.0,"black");

context.fillStyle = linearGrad;

context.fillRect(0,0,800,800);

window.onload = function(){

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

canvas.width = 1200;

canvas.height = 800;

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

var skyStyle = context.createLinearGradient(0,0,0,canvas.height);

skyStyle.addColorStop(0.0,'black');

skyStyle.addColorStop(1.0,'#035');

context.fillStyle = skyStyle;

context.fillRect(0,0,canvas.width,canvas.height);

for(var i=0;i<200;i++){

var r = Math.random()\*5+5;

var x = Math.random()\*canvas.width;

var y = Math.random()\*canvas.height\*0.65;

var a = Math.random()\*360;

drawStart(context,x,y,r,a);

}

fillMoon(context,2,900,200,100,30);

drawLand(context);

}

function drawLand(cxt){

cxt.save();

cxt.beginPath();

cxt.moveTo(0,600);

cxt.bezierCurveTo(540,400,600,800,1200,600);

cxt.lineTo(1200,800);

cxt.lineTo(0,800);

cxt.closePath();

var landStyle=cxt.createLinearGradient(0,800,0,0);

landStyle.addColorStop(0.0,"#030");

landStyle.addColorStop(1.0,'#580');

cxt.fillStyle = landStyle;

cxt.fill();

cxt.restore();

}

function drawStart (cxt,x,y,R,rot) {

cxt.save();

cxt.translate(x,y);

cxt.rotate(rot/180\*Math.PI);

cxt.scale(R,R);

startPath(cxt);//标准的五角星

cxt.fillStyle = "#fb3";

//cxt.strokeStyle = "#fd5";

//cxt.lineWidth = 3;

//cxt.lineJoin = "round"

cxt.fill();

//cxt.stroke();

cxt.restore();

}

function startPath(cxt){

cxt.beginPath();

for(var i=0;i<5;i++){

cxt.lineTo(Math.cos((18+i\*72)/180\*Math.PI),

-Math.sin((18+i\*72)/180\*Math.PI));

cxt.lineTo(Math.cos((54+i\*72)/180\*Math.PI)\*0.5,

-Math.sin((54+i\*72)/180\*Math.PI)\*0.5);

}

cxt.closePath();

}

function fillMoon(cxt,d,x,y,R,rot,fillColor){

cxt.save();

cxt.translate(x,y);

cxt.rotate(rot\*Math.PI/180);

cxt.scale(R,R);

pathMoon(cxt,d);

cxt.fillStyle = fillColor||"#fb5";

cxt.fill();

cxt.restore();

}

function pathMoon(cxt,d){

cxt.beginPath();

cxt.arc(0,0,1,0.5\*Math.PI,1.5\*Math.PI,true);

cxt.moveTo(0,-1);

cxt.quadraticCurveTo(1.2,0,0,1);

cxt.closePath();

}

function dist(x1,y1,x2,y2){

return Math.sqrt((x1-x2)\*(x1-x2)+(y1-y2)\*(y1-y2));

}

</script>

### 小结，曲线的绘制

