目录

[一.简介 2](#_Toc504050662)

[1.1 什么是canvas 2](#_Toc504050663)

[1.2 主要应用的领域（了解） 2](#_Toc504050664)

[二.canvas绘图基础 2](#_Toc504050665)

[2.1 canvas标签 2](#_Toc504050666)

[2.2 canvas绘图上下文context 3](#_Toc504050667)

[2.2.1 context：canvas的上下文、绘制环境。 3](#_Toc504050668)

[2.2.2 绘制表格 3](#_Toc504050669)

[2.2.3 beginPath（） 4](#_Toc504050670)

[2.2.4 清除矩形clearRect（x,y,w,h） 4](#_Toc504050671)

[2.3 绘制图片 4](#_Toc504050672)

[2.3.1 基本绘制图片的方式 4](#_Toc504050673)

[2.3.2 在画布上绘制图像，并规定图像的宽度和高度 4](#_Toc504050674)

[2.3.3裁切图片 4](#_Toc504050675)

[2.3.4 js创建img对象 5](#_Toc504050676)

[2.4 序列帧动画（与绘制图片合用） 5](#_Toc504050677)

# 一.简介

## 1.1 什么是canvas

1）canvas是一个矩形区域的画布，可以用JS在上面绘画，控制其每一个象素

2）canvas标签使用JS在网页上绘制图像，本身不具备绘图功能

3）canvas拥有多种绘制路径，矩形、圆形、字符以及添加图像的方法

4）HTML5之前的web页面只能用一些固定样式的标签：比如p、div、h1等，但有了canvas web页面可以丰富多彩。

## 1.2 主要应用的领域（了解）

1）游戏：canvas在基于web的图像显示方面比Flash更加立体、更加小巧，canvas游戏流畅度和跨平台方面更牛。

2）可视化数据：数据图表示

3）banner广告：Flash曾经辉煌的时代，智能手机还未曾出现，现在以及未来的智能时代，HTML5技术能够在banner广告上发挥巨大作用，用canvas实现动态的广告效果再合适不过。

4）未来=>模拟器：无论从视觉效果还是核心功能方面来说，模拟器产品可以完全由Javascript来实现。

5）未来=>远程计算机控制：canvas可以让开发者更好的实现基于Web的数据传输，构建一个完美的可视化控制界面

6）未来=>图形编辑器：Photoshop图形编辑器能够100%基于web实现

# 二.canvas绘图基础

## 2.1 canvas标签

可以设置width和height属性

设置canvas标签的宽高是通过canvas标签的属性进行设置，不要用css去设置，如果不支持canvas的浏览器，会将canvas标签当做div，canvas标签之内的内容会显示出来

## 2.2 canvas绘图上下文context

### 2.2.1 context：canvas的上下文、绘制环境。

上下文：是所有绘制功能的API集合

例：

<**canvas id="demo"**></**canvas**>

//1.拿到canvas标签

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

//2.拿到context上下文

**var** ctx = canvas.getContext**("2d")**;//3d是webGL

//3.绘制

### 2.2.2 绘制表格

**var** canvas = **document**.getElementById**("demo")**;  
**var** ctx = canvas.getContext**("2d")**;  
**var** rectH = 10;  
**var** rectW = 10;  
ctx.**lineWidth** = .5;  
canvas.**style**.**border** = **"2px solid #000"**;  
**for(var** i=0; i < canvas.**width**/rectH; i++**)**{  
 //横线  
 ctx.moveTo**(**0, i \* rectH**)**;  
 ctx.lineTo**(**canvas.**width**, i \* rectH**)**;  
 //竖线  
 ctx.moveTo**(**i \* rectW, 0**)**;  
 ctx.lineTo**(**i \* rectW, canvas.**height)**;  
}  
ctx.stroke**()**;

### 2.2.3 beginPath（）

开始一个新状态，即重新开始一个路径

### 2.2.4 清除矩形clearRect（x,y,w,h）

清除矩形内的内容，相当于画图工具中的橡皮擦

## 2.3 绘制图片

### 2.3.1 基本绘制图片的方式

context.drawImage（img，x，y）

参数说明：x，y绘制图片左上角的坐标，img是绘制图片的dom对象

### 2.3.2 在画布上绘制图像，并规定图像的宽度和高度

content.drawImage(img，x，y, width, height);

参数说明：width绘制之图片的宽度，height绘制图片的高度

如果指定宽高，最好成比例，不然图片会被拉伸

等比公式： toH=height \* toW / width

设置高=原高度\*设置宽/原宽度

### 2.3.3裁切图片

content.drawImage(img，sx，sy, swidth, sheight, x, y, width, height);

参数说明：

sx， sy 裁剪的左上角坐标

swidth, sheight：裁剪图片的宽高

### 2.3.4 js创建img对象

**var** img = **new Image()**;  
img.**src** = **""**;  
img.onload = **function()**{}

## 2.4 序列帧动画（与绘制图片合用）

img.onload = **function()**{  
 **var** frameIndex = 0;//此变量标志序列，frame是“帧”的意思  
 setInterval**(function()**{  
 ctx.clearRect**(**0, 0, canvas.**width**, canvas.**height)**;//清除加载上一帧的痕迹  
 ctx.drawImage**(** img,  
 frameIndex \* 40,//表示每一帧原始位置，40是每一帧的宽度  
 sy,  
 swidth,  
 sheight,  
 x,  
 y,  
 **width**,  
 height**)**;  
 frameIndex++;  
 }, 1000/10**)**;//1000/10的意思是一秒绘制10帧（1s绘制10张照片）