html5Canvas基础

Canvas是什么？

- Canvas就是画布
- Canvas元素用于在网页上绘制2D图形和图像

Canvas使用场景有哪些？

- 动画
- H5游戏
- 图表

学习canvas之前需要掌握的知识

- HTML
- CSS
- JavaScript

html5Canvas基础——知识概要

- 使用Canvas画直线、矩形、圆形以及设置它们的样式
- Canvas中的图形变换、渐变、文字和图片
- Canvas的像素获取、阴影和曲线绘制以及区域的剪辑
- Canvas动画、交互和离屏技术

html5Canvas基础——实例

个性化名片

使用Canvas画基本图形

- Canvas的坐标体系
- 使用Canvas画直线、矩形、圆形
- 为图形设置样式

Canvas坐标体系

- canvas默认大小：300x150
- 通过HTML、CSS、JavaScript设置width和height的区别
  - HTML和JavaScript设置的是画布大小
  - CSS设置的是画布缩放后的大小
- 坐标系原点及方向
  - 原点在左上角，向右为x方向，向下为y方向

画直线、矩形和圆形

- 画直线：ctx.moveTo(x1, y1)， ctx.lineTo(x2, y2)
- 画圆形： ctx.arc(x, y, radius, 0, Math.PI*2, true)
- 画矩形：可以通过直线来画，也可以直接用 ctx.strokeRect(x1, y1, x2, y2)

beginPath和closePath

- beginPath和closePath并不是成对出现的
- beginPath的作用是开始一条新路径
- closePath的作用是使当前路径闭合

描边与填充样式

- strokeStyle用来设置画笔样式，也就是直线、曲线、边框的样式
- fillStyle用来设置画刷的样式，也就是填充样式
- lineWidth设置线条的粗细

注意：这里的样式，是指颜色、渐变色等

Canvas中的图形变换、渐变、文字和图片

Canvas中的图形变换

Canvas中的渐变

Canvas中的文字

Canvas中的图片

Canvas中的图形变换

图形变换都是针对坐标系而言

√ 平移：ctx.translate(x, y)

√ 旋转：ctx.rotate(rad)

√ 缩放：ctx.scale(x, y);

save和restore

用来保存和恢复上下文的环境ctx，一般成对出现

√ ctx.save()：保存当前上下文环境；

√ ctx.restore()：恢复到上一次的上下文环境。

Canvas中的渐变

- 线性渐变：ctx.createLinearGradient(xStart, yStart, xEnd, yEnd);
  - (xStart, yStart)是线段起点，(xEnd, yEnd)是线段终点
  - 起点到终点之间的颜色呈渐变
- gradient.addColorStop可以来控制渐变的颜色
- 渐变可以理解为一种颜色

Canvas中的渐变（续）

- **径向渐变：**

  ctx.createRadialGradient(xStart,yStart,radiusStart,xEnd,yEnd,radiusEnd);

  - **(xStart,yStart)是第一个圆的圆心，radiusStart是第一个圆的半径，(xEnd,yEnd)是第二个圆的圆心，radiusEnd是第二个圆的半径**
  - **第一个圆到第二个圆之间的颜色呈渐变**

Canvas中的文字

- 描边文字：ctx.strokeText(text, x, y)
- 填充文字：ctx.fillText(text, x, y);
- 设置字体样式：ctx.font
  - 例如ctx.font = "bold 100px sans-serif";
- 设置水平对齐方式：ctx.textAlign
  - left, start，左对齐；center，居中对齐；end,right，右对齐

Canvas中的文字（续）

- 设置垂直对齐方式：ctx. textBaseline
  - top，顶对齐；middle，居中；bottom，底部对齐
- 计算文本宽度：ctx.measureText(text).width
  - 须在设置字体样式之后计算

Canvas图片

- **绘制图片3种方法**
  - **ctx.drawImage(image,x,y)，该方法把图片绘制在(x,y)处**
  - **ctx.drawImage(image,x,y,w,h)，该方法把图片绘制在(x,y)处，并缩放为宽w，高h**
  - **ctx.drawImage(image,sx,sy,sw,sh,dx,dy,dw,dh)，该方法把图片中(sx,sy)处的宽sw，高sh的区域，绘制到(dx,dy)处，并缩放为宽dw，高dh**

Canvas图片（续）

在Image加载完成之后绘制

- 例：

```
var img = new Image();
img.src = "logo.png";
img.onload = function () {
    ctx.drawImage(img, 0, 0, 40, 40, 0, 0, 80, 80);
}
```

Canvas绘制

- Canvas的图形画刷和像素获取
- Canvas阴影绘制
- Canvas剪辑区域
- Canvas曲线绘制

Canvas图形画刷

- ctx.createPattern可以创建一个画刷模式，进而可以设置到fillStyle里，进行画刷的填充
- 函数原型：ctx.createPattern(image, type)

type取值：no-repeat：不平铺

repeat-x：横方向平

repeat-y：纵方向平铺

repeat：全方向平铺

Canvas像素操作

- 获取像素：var imageData = ctx.getImageData(x, y, w, h)
  - 返回的是一维数组，[r1,g1,b1,a1,r2,g2,b2,a2...]
- 设置像素：ctx.putImageData(imageData, x, y)
  - 把imageData放在(x, y)处
- 设置像素：ctx.putImageData(imageData, x, y, dirtyX, dirtyY, dirtyW, dirtyH)
  - 只显示(dirtyX, dirtyY)处的宽dirtyW，高dirtyH的区域

Canvas阴影绘制

- ctx.shadowOffsetX：阴影x方向的偏移距离
- ctx.shadowOffsetY：阴影y方向的偏移距离
- ctx.shadowColor：阴影的颜色
- ctx.shadowBlur：阴影的模糊半径

Canvas剪辑区域

第一步：设置一个路径

第二步：调用ctx.clip()

第三步：再绘制图形

Canvas绘制曲线

弧线：

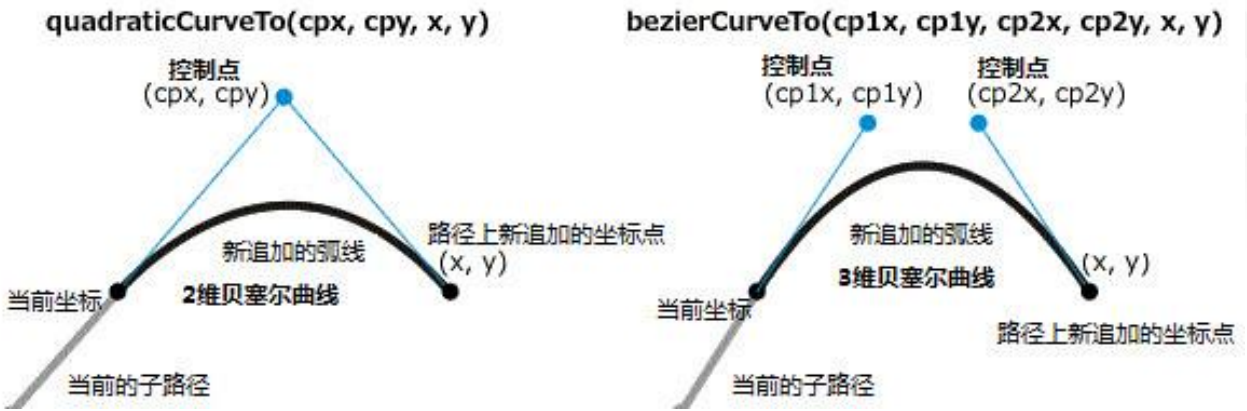
context.arc(x, y, radius, starAngle,endAngle, anticlockwise)

- 圆心(x, y)，半径radius
- 从starAngle到endAngle
- anticlockwise代表是否逆时针方向

Canvas绘制曲线（续）

- **二次样条曲线：context.quadraticCurveTo(qcpx,qcpy,qx,qy)**
- **贝塞尔曲线：**

  context.bezierCurveTo(cp1x,cp1y,cp2x,cp2y,x,y)



Canvas绘制曲线生成工具

二次贝塞尔曲线：

http://blogs.sitepointstatic.com/examples/tech/canvas-curves/quadratic-curve.html

三次贝塞尔曲线：

http://blogs.sitepointstatic.com/examples/tech/canvas-curves/bezier-curve.html

Canvas动画和离屏技术

- Canva动画
- Canvas离屏技术

Canvas动画

- **ctx.clearRect(x, y, width, height)**
  - **清除(x,y)点起，宽width，高height的区域，用于重新绘制**

Canvas离屏技术（续）

- **离屏技术是什么？**
  - 通过在离屏Canvas中绘制元素，再复制到显示Canvas中，从而大幅提高性能的一种技术。
- **什么时候使用离屏技术？**
  - 静态场景绘制特别耗资源，动态场景绘制简单。为了不每次更新动态场景的时候，都去绘制静态场景。
  - 一般把静态场景绘制在离屏canvas上，更新动态场景的时候，把静态场景copy过来，而不是重新绘制。

Canvas离屏技术（续）

- **一个Canvas中的图像绘制到另一个Canvas三种方法：**
  - **ctx.drawImage(canvas,x,y)，该方法把canvas绘制在(x,y)处**
  - **ctx.drawImage(canvas,x,y,w,h)，该方法把canvas绘制在(x,y)处，并缩放为宽w，高h**
  - **ctx.drawImage(canvas,sx,sy,sw,sh,dx,dy,dw,dh)，该方法把canvas中(sx,sy)处的宽sw，高sh的区域，绘制到(dx,dy)处，并缩放为宽dw，高dh**

总结

- Canvas能做什么
  - 画图，文字，阴影，变换，动画，图像处理……
- 学习Canvas的方法
  - 不要死记硬背，抓住主线，学会自己找资料解决问题