复习：

HTML5新特性 —— 十个

(1)新的语义标签

(2)表单2.0

(3)视频和音频

<video src="x.mp4" autoplay controls loop>

<audio src="x.mp3" autoplay controls loop>

(4)Canvas绘图

<canvas width="500" height="400"></canvas>

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

console.dir(ctx)

(5)SVG绘图

(6)拖放API

(7)地理定位

(8)WebWorker

(9)WebStorage

(10)WebSocket

1.使用Canvas绘制图形 —— 矩形

定位点：矩形左上角

ctx.lineWidth = 1

ctx.strokeStyle = '#000'

ctx.fillStyle = '#000'

ctx.strokeRect(x, y, w, h)

ctx.fillRect(x, y, w, h)

ctx.clearRect(x, y, w, h)

2.使用Canvas绘制图形 —— 文本

定位点：在文本基线的最左侧

ctx.textBaseline = 'alphabetic'

ctx.font = '12px sans-serif'

ctx.fillText(txt, x, y)

ctx.strokeText(txt, x, y)

ctx.measureText(txt).width

3.使用渐变对象绘制图形

1)创建渐变对象必须指定用在画布上的起点和终点坐标：

var g = ctx.createLinearGradient(x1,y1,x2,y2);

2)为渐变对象添加两个以上的颜色点

g.addColorStop( offset, color );

3)绘制图形时应用渐变对象

ctx.fillStyle = g;

ctx.strokeStyle = g;

4.使用Canvas绘制图形 —— 绘制路径(Path)

提示：Canvas中的路径概念与PS中的钢笔工具的含义一样的。可以绘制任意的直线、折线、曲线。路径本身是不可见的，有三个用途：描边、填充、裁剪

ctx.beginPath( ) 开始一条新路径

ctx.closePath( ) 闭合路径(自动连接终点和起点)

ctx.moveTo(x, y) 移动到指定点

ctx.lineTo(x, y) 到指定点绘制直线

ctx.arc(x, y, r, startAngle, endAngle) 绘制圆拱路径

ctx.ellipse(x, y, rx, ry, startAngle, endAngle) 绘制椭圆拱路径

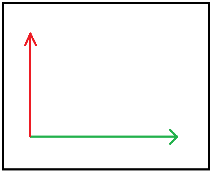
ctx.bezierCurveTo(....) 绘制贝塞尔曲线

ctx.stroke( ) 对当前路径进行描边

ctx.fill( ) 对当前路径进行填充

ctx.clip( ) 对当前路径进行裁切

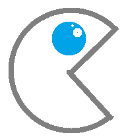
练习：使用直线路径绘制一个坐标轴



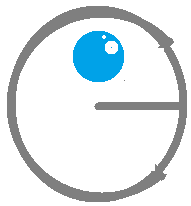
练习：使用圆拱绘制一个不断前进的进度条，从-90度开始前进



练习：创建一个函数openMouth(),使用圆拱绘制如下图形



练习：创建一个函数closeMouth(),使用圆拱绘制如下图形



练习：使用定时器，每隔500ms，分别调用openMouth()和closeMouth()

5.使用Canvas绘制图形 —— 绘制图像

提示：图像的定位点在自己的左上角。必须等待服务器端的图片下载完成，才能开始绘制图像！

var img = new Image();

img.src = 'x.png'; //浏览器会立即异步请求该图片

img.onload = function(){ //图片加载完成事件

ctx.drawImage(img, x, y) //原始大小绘制图像

ctx.drawImage(img, x, y, w, h) //绘制拉伸后的图像

}

练习：

(1)在500\*400画布的左上角绘制一幅飞机

(2)在500\*400画布的右上角绘制一幅飞机

(3)在500\*400画布的左下角绘制一幅飞机

(4)在500\*400画布的右下角绘制一幅飞机

(5)在500\*400画布的中央绘制一幅2倍大小的飞机

(6)在500\*400画布上绘制一幅从左到右移动的飞机

(7)在500\*400画布绘制一幅可以随鼠标而移动的飞机

var x = 0;

var y = 0;

canvas.onmousemove = function(e){

x = e.offsetX;

y = e.offsetY;

}

setInterval(....{ x, y }, 50)

6.图形/图像的变形问题 —— 难点

Canvas绘图中一般不使用CSS的变形属性——CSS变形只能用于HTML元素-Canvas技术中只有一个<canvas>。

若只针对某个图形/图像做变形，需要使用画笔的变形相关方法：

ctx.rotate( deg ) 旋转画笔，所绘的图形图像都会旋转

ctx.scale( x, y ) 缩放

ctx.translate( x, y ) 平移整个画布的原点到一个新的点

ctx.save() 保存画笔的当前变形状态

ctx.restore() 恢复最近一次保存的画笔的所有状态

注意：

(1)画笔旋转默认的轴点是画布的原点(0,0)，而不是图形/图像的定位点

(2)画笔的旋转有累加效果

(3)可以通过ctx.translate()修改画布原点的位置，一旦修改，所有点的坐标值全部改变

练习(有难度)：在一张画布上绘制三个小飞机，飞机1在左上角以自己为中心原地旋转；飞机2在右上角不旋转；飞机3在左下角以自己为中心原地旋转

提示：每个飞机平移+旋转+绘图完成后，必须逆向旋转+逆向平移——恢复画笔最开始的状态，才不会对下一个飞机造成影响

平移 => 旋转 => 绘图 => 逆向旋转 => 逆向平移

7.扩展小知识：第三方提供的绘制统计图的工具

(1)Chart.js

(2)FusionCharts

(3)EChart.js

注意：学习第三方工具的重点在于过程，不在于记忆！

**自学一个第三方工具的过程：**

(1)查找官网，查看官方说明和介绍

http://www.chartjs.org/

Simple yet flexible JavaScript charting for designers & developers。Open source、8 Chart types、HTML5 Canvas、Responsive。

(2)仿写示例程序

new Chart(c2, {

type: 'bar',

data: {},

options: {}

});

(3)在自己的项目中应用该工具

课后练习：

1. 使用Canvas绘制一个随机改变的验证码图片



var str = 'ABCDEFGHJKLMNPQRSTWXY3456789';

var char = str[ 0~字符串长度间的随机数 ];

要求：

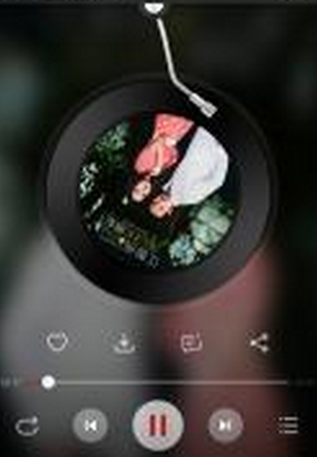
画布背景颜色随机(浅色) ctx.fillRect()

文字内容随机、大小随机，颜色随机(深色)、旋转角度随机。

5条随机干扰线（深色），处于文字上方。

100个杂色点（半径为1为圆），处于文字上方。

2.仿网易云音乐的播放界面



点击播放按钮，碟片开始旋转，背景音乐开始播放；

再次点击播放按钮，碟片停止旋转，背景音乐停止播放。

难点提示：如何为“按钮”绑定事件监听？该按钮是不是标签，是ctx.drawImage()绘制的图像，只能为CANVAS绑定事件监听，计算点击的坐标是否处于“按钮”图片范围内部。