复习：

HTML5新特性：

(1)新的语义标签和属性

(2)表单2.0

(3)视频和音频

<video src="" autoplay loop controls muted poster preload="auto">

<audio src="">

(4)Canvas绘图 —— 重点

(5)SVG绘图

(6)地理定位

(7)拖放API

(8)WebWorker

(9)WebStorage

(10)WebSocket

JS绘图技术：

<canvas width="600" height="400"></canvas>

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

//常用属性

ctx.fillStyle = 颜色/渐变对象

ctx.strokeStyle =颜色/渐变对象

ctx.lineWidth = 1

ctx.font = '10px sans-serif'

ctx.textBaseline = 'alphabetic'

ctx.shadowColor = 'rgba(0,0,0,0)'

ctx.shadowOffsetX = 0

ctx.shadowOffsetY = 0

ctx.shadowBlur = 0

//常用方法

(1)绘制矩形

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

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

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

(2)绘制文本

ctx.fillText(txt, x, y)

ctx.strokeText(txt, x, y)

ctx.measureText(txt).width

(3)绘制路径

(4)绘制图像

今日目标：

(1)使用Canvas绘制路径和图像 —— 重点&难点

(2)使用第三方绘图工具 —— Chart.js，掌握

1.使用Canvas绘制路径(Path)

提示：Canvas中的路径概念与Photoshop中钢笔工具类似的。路径本身是不可见的，有三个用途：描边、填充(闭合)、裁剪(闭合)

Canavs中与路径绘制相关方法：

ctx.beginPath() //开始一条新路径

ctx.closePath() //闭合路径，让最后一个锚点自动连接到第一个锚点

ctx.moveTo(x, y) //移动到指定点

ctx.lineTo(x,y) //从当前点到指定点绘制直线路径

ctx.arc() //绘制拱形路径

ctx.ellipse() //绘制椭圆路径

ctx.bezierCurveTo() //绘制贝塞尔曲线路径

----------------------------------------------

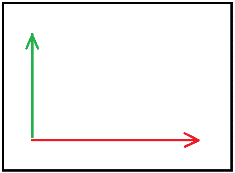
ctx.stroke()

ctx.fill()

ctx.clip()

练习：

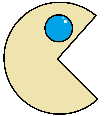
(1)使用直线路径绘制坐标轴



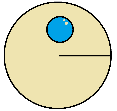
(2)使用圆拱+定时器绘制可以前进的进度条

(3)创建一个函数：openMouth()，在画布上绘制如下的图形：



(4)创建一个函数：closeMouth()，在画布上绘制如下的图形：



(5)使用定时器，不停的调用openMouth()和closeMouth()

2.使用Canvas绘制图像

提示：图片的定位点在图片的左上角。客户端JS必须等待图片加载完成才能开始绘制。

var img = new Image();

img.src = 'x.jpg';

img.onload = function(){

//图片已经加载完成了

**ctx.drawImage(img, x, y) //使用默认的宽高**

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

}

练习：

(1)在画布的四个角各绘制一个小飞机

(2)在画布的中央绘制一个2倍标准大小的飞机

(3)\*绘制一个可以随着鼠标而移动的小飞机，提示需要使用mousemove

注意：Canvas绘图中，只有一个HTML元素——canvas！！其它图形图像都不是元素，不能绑定事件函数！！

(4)\*绘制一个可以左右移动的小飞机

(5)\*\*绘制一个可以在画布范围内走斜线移动的小飞机——碰到画布的任何一个边缘，立即反弹

|  |
| --- |
| 小结：Canvas绘图可以绘制的内容：  (1)矩形： ctx.fillRect() ctx.strokeRect() ctx.clearRect()  (2)文本： ctx.fillText() ctx.strokeText() ctx.measureText()  (3)路径 - 描边/填充/裁剪  ctx.beginPath() ctx.closePath()  ctx.moveTo() ctx.lineTo() ctx.arc()  ctx.stroke() ctx.fill() ctx.clip()  (4)图像： ctx.drawImage()  (5) |

3.绘图上下文的状态改变和恢复 —— 难点&晦涩

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

//可以将绘图上下文对象（即画笔对象）进行变形(transform)——与对Canvas施加CSS Transform样式不同，绘图上下文的变形只影响当前绘制的图形图像内容

ctx.translate(x,y) //坐标轴原点平移到指定点，所有点的坐标都发生改变

ctx.rotate(deg) //画笔旋转，则内容旋转，轴点在坐标轴原点

ctx.scale() //画笔缩放

====================

ctx.save() //保存绘图上下文(画笔）当前的变形数据

ctx.restore() //恢复最近一次保存的画笔的变形相关的状态

练习：\*\*\*有点坑——绘制四个小飞机，各在画布的一个角在绕着自己的中心在旋转。

项目中Canvas技术的主要用途：

(1)绘制统计图

(2)小游戏

(3)绘图板

(4)动态的背景（带交互带动画）

4.使用第三方统计图绘制工具——Chart.js——重点在于自学过程

提示：第三方的绘图工具非常多！直接百度“JS绘图工具”！

第三方工具的使用步骤：

**(1)打开官网，看工具介绍**

http://www.chartjs.org/

Simple yet flexible JavaScript charting for designers & developers。一款开源的、提供了8中图表的、基于Canvas、响应式图表绘制工具库。

**(2)参考DEMO，编写示例程序**

new Chart(canvasId, {

type: 'bar', //图表的类型，共8中

data: { }, //图表必需的数据

options: { } //可选项

});

**(3)查看API Document，实现自己的项目需求**

参考手册中的示例代码

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：

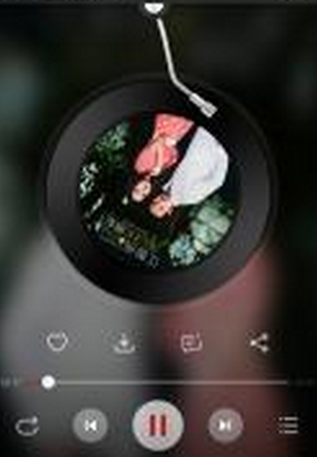
画布背景颜色随机(浅色) ctx.fillRect()

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。