复习：

H5新特性——10个

(1)新的语义标签

(2)增强型表单

新的input type：

新的表单元素：

datalist、progress、meter、output

表单元素的新属性：

autofocus、placeholder、multiple、form、

required、minlength、maxlength、min、max、pattern

(3)视频和音频

<video src="x.mp4" autoplay controls loop muted poster="x.png" preload="auto"></video>

<audio src="x.mp3" autoplay controls loop muted preload="auto"></video>

(4)Canvas绘图

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

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

console.dir(ctx)

(5)SVG绘图

(6)地理定位

(7)拖放API

(8)Web Worker

(9)Web Storage

(10)WebSocket

练习：

使用Node.js + Express模块创建一个Web服务器，可以向客户端提供两个静态URI：

**/stat.html**  提供一个空白的Canvas

**/js/jquery-1.11.3.js**

外加一个动态数据URI：

**GET /stat**  向客户端返回如下的JSON数据：

[

{"label": "部门1", "value":300},

{"label": "部门2", "value":500},

{"label": "部门3", "value":150},

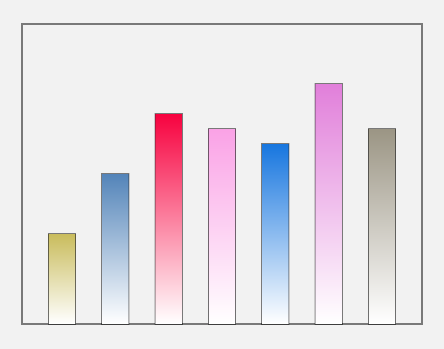
{"label": "部门4", "value":400},

{"label": "部门5", "value":550},

{"label": "部门6", "value":250}

]

根据这些数据，绘制出如下图所示的统计图：



提示：为简化起见，可以把value值看做每个柱的高度。

今日话题：

使用Canvas绘制文本

使用Canvas绘制路径

使用Canvas绘制图像

使用Canvas绘制变形和动画 —— 难点

1.补充：Canvas的尺寸如何设置

Canvas画布的尺寸若用CSS来指定，并未修改真正的尺寸，而是进行“拉伸”—— Canvas尺寸不能用CSS指定！！

可以使用HTML元素的属性、或JS对象属性来指定！

2.使用Canvas的画笔绘制矩形

ctx.fillStyle = '#000'

ctx.strokeStyle = '#000'

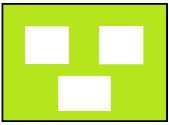
ctx.lineWidth = 1 描边/线条的宽度

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

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

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

练习：使用矩形进行抠图



|  |
| --- |
| 扩展小知识：在绘图时使用渐变对象  **线性渐变（linearGradient）：**  var g = ctx.createLinearGradient(x1,y1, x2, y2);  g.addColorStop( offset, color )  .... //渐变对象中至少有两个颜色点  ctx.fillStyle = g; //将渐变对象用于填充  ctx.strokeStyle = g; //将渐变对象用于描边  **径向渐变（radialGradient）** |

练习：在页面中绘制三个柱子，每个柱子都是从一个随机颜色到白色的渐变填充色。

|  |
| --- |
| 扩展小知识：Canvas绘图时实现动画 —— 网页游戏  原理：使用周期性定时器执行“清画布，再重新绘制内容” |

练习：

绘制一个矩形，从左上角到左下角移动，再弹回；

绘制一个矩形，从左上角到右下角45度角移动，遇到边界就弹回；

绘制一个矩形，从左上角到右下角30度角移动，遇到边界就弹回；

var x = 0;

var xDirection = 1; //1表右行 -1表左行

var y = 0;

setInterval(function(){

ctx.clearRect(0, 0, 500, 400); //清除已有内容

ctx.strokeRect(x, y, 100, 80); //重新绘制内容

x += 10\*xDirection;

if(x>=400){ //控制前进方向

xDirection = -1;

}else if(x<=0){

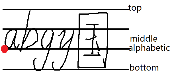
xDirection = 1;

}

}, 50);

3.使用Canvas的画笔绘制文本

文本的定位点在文本基线的起点



ctx.font = '10px sans-serif' 设置字体及大小

ctx.textBaseline = 'alphabetic' 设置文本基线

ctx.fillText( txt, x, y ) 填充文本

ctx.strokeText( txt, x, y ) 描边文本

ctx.measureText( txt ).width 基于当前字体测量指定文本的宽度

练习： 在画布左上角/右上角描边一段文本

在画布左下角/右下角填充一段文本

实现一个文本跑马灯效果



4.使用Canvas的画笔绘制路径

Path：路径，类似于Photoshop中的“钢笔工具”，路径本身不可见，可用于：(1)描边 (2)填充 (3)选区裁剪

ctx.beginPath() 开始一条新的路径

ctx.closePath() 闭合路径

ctx.moveTo(x, y) 移动到某一点

ctx.lineTo(x, y) 从当前点到指定点绘制直线

ctx.arc(cx, cy, r, start,end ) 绘制圆形/圆弧路径

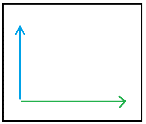
ctx.ellipse( cx,cy,rx,ry,start,end ) 绘制椭圆路径

ctx.stroke() 使用当前路径进行描边

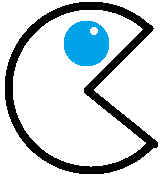
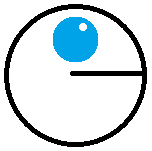
ctx.fill() 使用当前路径进行填充

ctx.clip() 使用当前路径进行裁剪

练习：使用路径绘制一个坐标轴



练习：使用圆形路径实现一个环形进度条

5.使用Canvas的画笔绘制图像

绘制图像时，客户端浏览器必须等待异步图片加载完成

var img = new Image()

img.src = "x.png"; //浏览器会自动去异步下载此图片

console.log(img.width); //0

img.onload = function(){ //图片加载完成事件

console.log(img.width); //有值

ctx.drawImage( img, x, y ) 原尺寸绘图

ctx.drawImage( img, x, y, w, h ) 缩放绘图

}

练习： 1在画布的左下角/右下角绘制一个原尺寸的飞机

2在画布的正中央绘制400\*200尺寸的飞机

3绘制一个从左向右移动的飞机

4绘制一个可以随鼠标而移动的飞机

var x = 0;

var y = 0;

canvas.onmousemove=function(e){

e.offsetX e.offsetY

}

setInterval(....)

课后练习：

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)。

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。