复习：

HTML5表单新特性：

(1)新的input type

text、password、radio、checkbox、button、submit、reset、image、file、hidden

url、email、number、tel、search、range、color、date、month、week

(2)新的表单元素

input、textarea、select/option、button

datalist、progress、meter、output

(3)表单元素的新属性

placeholder、autocomplete、autofocus、form、multiple

max、min、step、minlength、maxlength、、pattern、required

今日目标：

(1)HTML5新特性之二——视频和音频——较重要

(2)HTML5新特性之三——Canvas绘图——最重要&难点

“页游”逐步取代“端游”

HTML5之前播放视频和音频使用Flash技术。

1.HTML5新特性——视频

<video src="x.mp4"></video>

常用属性：

autoplay: false 是否自动播放

controls:false 是否显示播放控件

muted:false 是否静音

loop:false 是否循环播放

currentTime:0 当前播放时间

duration:46.56 总时长

paused:fasle 当前是否处于暂停状态

playbackRate:1 回放速率，如0.5或2...

poster:'' 播放第一帧之前显示的电影海报(图片)

volumn:1 音量，1代表100%

preload: auto 指定视频的预加载(影片缓存)方案：

auto：默认值，启用预加载，加载影片描述信息，以及预加载几秒钟的内容

metadata：只预加载影片的描述信息

none：不预加载任何内容

常用方法：

play: 开始播放视频

pause: 暂停播放视频

常用事件：

onplay: 视频开始播放事件

onpause: 视频开始暂停事件

练习：使用一张图片（播放&暂停），模拟一个按钮效果，点击后可以播放/暂停视频

练习：当视频开始暂停时，显示一个广告图片，视频开始播放时，隐藏广告图片

2.HTML5新特性——音频

<audio src="x.mp3"></audio>

常用属性：

autoplay: false 是否自动播放

controls:false 是否显示播放控件

muted:false 是否静音

loop:false 是否循环播放

currentTime:0 当前播放时间

duration:46.56 总时长

paused:fasle 当前是否处于暂停状态

playbackRate:1 回放速率，如0.5或2...

volumn:1 音量，1代表100%

preload: auto 指定视频的预加载(影片缓存)方案：

auto：默认值，启用预加载，加载影片描述信息，以及预加载几秒钟的内容

metadata：只预加载影片的描述信息

none：不预加载任何内容

常用方法：

play: 开始播放音频

pause: 暂停播放音频

常用事件：

onplay: 开始播放事件

onpause: 开始暂停事件

onend: 播放结束

练习：使用复选框实现背景音乐的暂停和播放



3.前端中的绘图技术

用途：网页画图板、走势图、统计图、地图、游戏等

种类：

(1)Canvas绘图

(2)SVG绘图

(3)WebGL绘图

4.Canvas绘图技术

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

您的浏览器不支持Canvas绘图技术！

</canvas>

行内块，默认300\*150px，可以使用width和height属性重定义画布的宽和高，不能使用CSS样式来重定义。

Canvas：画布，用于绘制二维的位图。在画布上绘图必须获得画布上的绘图上下文对象：

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

console.dir(ctx);

**绘图上下文对象的常用属性：**

1. fillStyle:"#000000" 填充样式
2. font:"10px sans-serif" 字体
3. globalAlpha:1 全局不透明度
4. lineCap:"butt"
5. lineJoin:"miter"
6. lineWidth:1 描边/轮廓宽度
7. shadowBlur:0
8. shadowColor:"rgba(0, 0, 0, 0)"
9. shadowOffsetX:0
10. shadowOffsetY:0
11. strokeStyle:"#000000" 描边/轮廓样式
12. textAlign:"start" 文本水平对齐
13. textBaseline:"alphabetic" 文本基线

**绘图上下文对象的常用方法：**

1. arc:*arc() 绘制拱形路径*
2. beginPath:*beginPath() 开始绘制路径*
3. bezierCurveTo:*bezierCurveTo() 绘制贝塞尔曲线*
4. clearRect:*clearRect() 清空矩形范围*
5. clip:*clip() 裁剪*
6. closePath:*closePath() 闭合路径*
7. createLinearGradient:*createLinearGradient()*
8. createRadialGradient:*createRadialGradient()*
9. drawImage:*drawImage() 绘制一幅图像*
10. ellipse:*ellipse() 绘制椭圆*
11. fill:*fill() 填充路径*
12. fillRect:*fillRect() 填充一个矩形*
13. fillText:*fillText() 填充一段文本*
14. lineTo:*lineTo() 绘制直线*
15. measureText:*measureText() 测量一段文本的宽度*
16. moveTo:*moveTo() 移动到指定点*
17. rect:*rect() 绘制矩形路径*
18. rotate:*rotate() 旋转*
19. scale:*scale() 缩放*
20. stroke:*stroke() 描边*
21. strokeRect:*strokeRect() 描边一个矩形*
22. strokeText:*strokeText() 描边一段文本*
23. translate:*translate() 平移*

|  |
| --- |
| 提示：Canvas学习的难点：  (1)单词量比较大，比较生僻  (2)大量数学坐标和尺寸运算 |

Content：内容

Context：上下文

4.使用绘图上下文对象绘制矩形——定位点在左上角

ctx.lineWidth = 1;

ctx.strokeStyle = "#000"

ctx.strokeRect(x,y,w,h) //描边一个矩形rectangle

ctx.fillStyle = "#000"

ctx.fillRect(x,y,w,h) //填充一个矩形

ctx.clearRect(x,y,w,h) //清空一个矩形范围

练习：在500x400画布上绘制100x80矩形，左上角描边一个，右上角描边一个，左下角填充一个，右下角填充一个，中央描边+填充一个

练习：创建一个可以上下移动的矩形 16:10

练习：创建一个可以斜向45度角移动的矩形

5.使用绘图上下文对象绘制圆形

提示：Canvas中不能直接绘制圆形和多边形——只能使用路径来绘制。

6.使用绘图上下文对象绘制文本——定位点在“基线”的最左侧

ctx.textBaseline='alphabetic';

ctx.measureText(txt).width //测量得到文本的总宽度

ctx.strokeStyle = '#000';

ctx.strokeText(txt, x, y) //对文本进行描边

ctx.fillStyle = '#000';

ctx.fillText(txt, x, y) //对文本进行填充

练习：在画布的左上角、右上角填充一段文本；在画布的左下角、右下角描边一段文本；在中央描边+填充一段文本

练习：在画布上绘制一段可以45度角移动的文本

|  |
| --- |
| 文本基线（text baseline）：    ctx.textBaseline可取值：  (1)'alphabetic'，默认值，字母基线，就是第三条线  (2)'top': 第一条线  (3)'bottom': 第四条线  (4)'middle': 第二三条线中间  提示：一般的编程语言中，习惯把文本左下角作为定位点，即习惯上textBaseline取值为bottom |

7.为图形和文本添加阴影特效

ctx.shadowOffsetX

ctx.shadowOffsetY

ctx.shadowColor

ctx.shadowBlur

8.为图形和文字设置渐变的颜色

渐变分为两种：

(1)线性渐变

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

g.addColorStop(0.1, '#f00'); //添加颜色点

g.addColorStop(0.9, '#0f0');

ctx.strokeStyle = '#f00' / g;

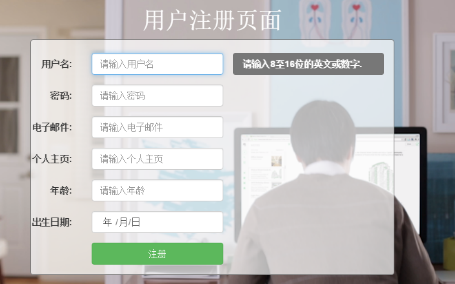
ctx.fillStyle = '#f00' / g;

(2)径向渐变

var g = ctx.createRadialGraident()

课后练习：

(1)使用视频做网页的背景



(2)假设使用AJAX从服务器端获取到如下的数据：

[

{"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":600},

{"label": "部门6", "value":250}

]

