复习：

HTML5新特性——10个

(1)新的语义标签

(2)表单2.0

新的input type——10

新的表单元素——4

表单元素的新属性——11

(3)视频和音频

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

<audio src="">

(4)Canvas绘图

(5)SVG绘图

(6)地理定位

(7)拖放API

(8)WebWorker

(9)WebStorage

(10)WebSocket

<canvas id="c1" width="500" height="400">

</canvas>

var ctx = c1.getContext('2d');

console.dir(ctx);

(1)绘制矩形

ctx.fillRect()

ctx.strokeRect()

ctx.clearRect()

(2)绘制文本

ctx.fillText()

ctx.strokeText()

ctx.measureText(txt).width

(3)绘制路径

ctx.beginPath()

ctx.closePath()

ctx.moveTo() ctx.lineTo()

ctx.arc() ctx.ellipse()

ctx.stroke() ctx.fill() ctx.clip()

(4)绘制图像

var img = new Image();

img.src = 'x.png';

img.onload = function(){

ctx.drawImage(img, x, y)

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

}

今日目标：

1)知识点拾漏

2)作业知识点

3)SVG绘图

1.绘图知识点：如何等待所有图片加载完成才开始绘图

浏览器加载图片属于“多线程异步加载”，完全无法预测哪一张图片先加载完成。但Canvas绘制图片时必须按一定顺序！——只能等待所有图片加载完成，才开始绘图！

var progress = 0; //总的加载进度

img1.onload = function(){

progress += 20; //加入当前图片的权重

if(progress===100){ startDraw(); }

}

img2.onload = function(){

progress += 80; //加入当前图片的权重

if(progress===100){ startDraw(); }

}

2.如何Canvas上的图形/图像绑定事件监听？

HTML中只能为元素/标签绑定监听函数；Canvas绘图中只有一个元素-CANVAS，每一个图形/图像都不是元素，不能直接进行事件绑定。解决办法：“事件委托”——让CANVAS监听所有的事件，计算事件发生坐标点，是否处于某个图形/图像的范围内。

对于标准几何图形可以进行事件绑定；

对于不标准几何图形进行事件绑定非常麻烦。

|  |
| --- |
| **位图：Photoshop,**由一个又一个颜色点组成，每个点由rgb指定。善于表现颜色的细节变化，放大后会失真——一般用于照片 —— Canvas绘图  **矢量图：Flash/AI,**由一个又一个矢量线条/图形组成，每个图形指定方向和颜色即可。可以无限放大而不失真，不善于表现颜色的细节变化——一般用于设计图标、动画 —— SVG绘图 |

3.HTML5新特性之五 —— SVG绘图

Scalable Vector Graphic：可缩放的矢量图；可以无限放大而不失真，不善于表现颜色的细节变化。

|  |  |  |
| --- | --- | --- |
| Canvas绘图和SVG绘图的比较 | | |
|  | **Canvas** | **SVG** |
| 出现时间 | 随着H5标准诞生 | 2000年即存在，后纳入H5标签 |
| 绘图类型 | 位图 | 矢量图 |
| 如何绘制 | 使用JS函数绘制 | 使用H5标签绘制 |
| 事件绑定 | 每个图形无法绑定监听 | 每个图形都可以绑定事件监听 |
| 使用场合 | 颜色细腻，如游戏、统计图 | 无限缩放，如图标、统计图、地图 |

**HTML5出现之前使用SVG的方法：**

(1)创建一个XML文档，编写SVG的标签

(2)在HTML引用XML文档，使用在IMG、IFRAME、EMBED、OBJECT等HTML标签中。

**HTML5出现之后使用SVG的方法：**

H5标准把SVG标签纳入自己的标签库（有些SVG的特效标签被舍弃了），所有SVG标签可以直接出现在H5文档中。注意：SVG标签中不能放置普通的HTML标签；SVG相关的标签页不能放在SVG画布外！

<svg>300\*150的inline-block</svg>

|  |
| --- |
| 思考题：  Canvas的尺寸不能使用CSS指定！  SVG画布的尺寸可以吗？？？ |

4.使用SVG绘图 —— 绘制矩形

<rect **width="" height=""** x="" y="" fill="" fill-opacity="" stroke="" stroke-width="" stroke-opacity=""></rect>

说明：

(1)所有的SVG标签样式可以使用属性来声明，也可以使用CSS形式来声明——但其可用的样式属性都是专有的！普通的CSS样式属性对SVG元素是无效。

(2) SVG标签属性都不是HTML DOM属性！可以使用核心DOM方式操作SVG标签属性——setAttribute()、getAttribute()

(3)动态的创建新的SVG元素，可以使用HTML字符串拼接方式，也可以使用document.createElementNS() / svg.appendChild()方式

练习：

(1)在画布的左上角绘制一个矩形，半透明的填充和半透明的边框，鼠标移动到上方时，填充和边框都变为不透明；鼠标离开恢复原样。

(2)在画布的左上角绘制一个矩形，从左到右在移动；

提示：定时器操作属性x(核心DOM操作)

(3)假设从服务器上获取如下的数据：

[ {"label": "部门1", "value":350},

{"label": "部门2", "value":450},

{"label": "部门3", "value":250} ]

根据如上数据绘制若干个矩形，作为统计图

提示：方法1-使用HTML字符串拼接 方法2-createElement()/appendChild()

document.createElementNS('http://www.w3.org/2000/svg','rect');

5.使用SVG绘图 —— 绘制圆形

<circle **r=""** cx="" cy="" fill="" stroke=" " stroke-width="" ....></circle>

练习：

(1)在画布上创建30个随机的圆形，位置随机、大小随机、填充颜色随机，透明度随机；点击某个圆形，渐渐的变大/变淡，直到fill-opacity<0.001把该元素从DOM树上删除

(2)选作：“蜻蜓点水”，点击画布某处，生成一个变淡变大的圆

6.使用SVG绘图 —— 绘制椭圆

<ellipse **rx="" ry=""** cx="" cy=""></ellipse>

7.使用SVG绘图 —— 绘制直线

<line **x1="" y1="" x2="" y="" stroke=""**></line>

提示：所有的SVG图形默认只有fill没有stroke；而line元素不需要fill只需要stroke！

练习：使用直线绘制如下的图标

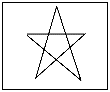


提示：若多个SVG图形有一样的属性，可以声明在<g></g>小组中，在小组上声明公共的属性，每个组员都会自动继承该属性。

8.使用SVG绘图 —— 绘制折线

<polyline **points="x1,y1 x2,y2 x3,y3 ..."** stroke="" fill="transparent"></polyline>

练习：绘制如下的图形



9.使用SVG绘图 —— 绘制多边形

<polygon **points="x1,y1 x2,y2 x3,y3 ..."**></polygon>

练习：使用两个多边形绘制如下的图标



10.使用SVG绘图 —— 绘制文本

<text alignment-baseline="before-edge" font-size="40" font-family="SimHei" x="50" y="50">abyg中国</text>

11.使用SVG绘图 —— 绘制图像

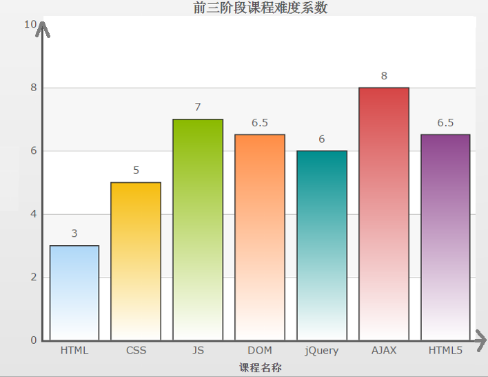
<image xlink:href="img/disc.png" width="300" height="50" x="100" y="100"></image>

课后练习：

1. 前端页面异步请求后台返回的如下JSON数据：

'[{"label":"HTML",value:3}, {"label":"CSS",value:5},....]'

根据这段JSON字符串，绘制下图：



渐变效果可以提前参考PPT，设法完成。

3)自学two.js第三方工具的使用，理解其作用，仿写官方示例代码，学会使用方法，实现如下效果：

