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

4.Canvas绘图

5.SVG绘图

6.地理定位

7.拖放API

8.WebWorker

9.WebStorage

10.WebSocket

Canvas绘图使用方法：

<canvas id="c" width="" height=""></canvas>

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

//绘制矩形

ctx.fillRect() ctx.strokeRect() ctx.clearRect()

//绘制文本

ctx.fillText() ctx.strokeText() ctx.measureText()

//绘制路径

ctx.beginPath() ctx.closePath()

ctx.moveTo() ctx.lineTo() ctx.arc()

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

//绘制图像

ctx.drawImage()

//图形图像变形

ctx.translate() ctx.rotate() ctx.scale()

ctx.save() ctx.restore()

今日目标：

(1)补充：Canvas小知识点

(2)SVG绘图技术 —— 重点

(3)第三方2D绘图工具 —— Two.js —— 了解

1.如何等待所有图片加载完成后，才开始绘图？

多张图片绘制需要按照特定的顺序，而加载完成顺序是完全无法预测，只能等待所有图片加载完成，才能开始绘制。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//典型的错误代码

var img1 = new Image();

img1.src = "";

img1.onload = function(){

ctx.drawImage(img1, x, y);

}

var img2 = new Image();

img2.src = "";

img2.onload = function(){

ctx.drawImage(img2, x, y);

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**var progress = 0;**  //加载进度 0~100

var img1 = new Image();

img1.src = "";

img1.onload = function(){

progress += 80;

if(progress===100){

startDraw();

}

}

var img2 = new Image();

img2.src = "";

img2.onload = function(){

progress += 20;

if(progress===100){

startDraw();

}

}

2.如何为Canvas上的图形/图像绑定事件监听？——难点

网页只能为DOM元素绑定监听函数，Canvas绘图技术中只要一个Canvas元素，其它图形图像都不是元素——无法进行事件绑定！——

解决办法：为Canvas绑定监听函数，获取事件发生的坐标，是否处于目标图形/图像所在范围内——只能为规则的图形“绑定”监听

|  |
| --- |
| **Adobe Photoshop**：处理位图——每幅图像由点(rgb)组成，善于描述颜色的细节变化，可用于照片等领域——放大后会出现马赛克失真。  **Adobe Illustrator**：处理矢量图——每幅图像由线条(需要指定方向、值)，可以无限缩放而不失真——不善于描述颜色的细节变化。 |

3.HTML5新特性之五——SVG绘图

|  |  |  |
| --- | --- | --- |
|  | **Canvas绘图** | **SVG绘图** |
| 绘图类型 | 位图 | 矢量图 |
| 缩放 | 失真 | 不失真 |
| 颜色细节 | 丰富 | 不够丰富 |
| 应用领域 | 照片、游戏 | 统计图、图标、地图 |
| 内容 | JS绘制 | 每个图形都是标签 |
| 事件绑定 | 不方便 | 方便 |

Scalable Vector Graphiph，可缩放的矢量图，此技术在2000年就已经存在了，独立于网页的一门技术；HTML5之后，纳入了HTML5标准标签库，并进行了一定的瘦身。

SVG技术的使用方法：

**(1)HTML5之前的使用方法：**

SVG标签不属于HTML4或XHTML标签，只能编写在独立的XML文件中，首先编写一个SVG文件（本质是一个XML文件），然后在HTML中使用IMG/IFRAME/OBJECT/EMBED等标签引入.svg文件

**(2)HTML5之后的使用方法——H5已经把SVG标签采纳：**

在HTML文件中直接使用SVG相关标签即可

<svg>默认为300\*150的inline-block</svg>

练习：

(1)使用SVG矩形绘制一个国际象棋的棋盘——使用HTML中的rect标签

(2)使用SVG矩形绘制一个国际象棋的棋盘——使用JS动态创建rect标签

4.使用SVG绘制矩形

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

注意：

(1)SVG图形的样式可以用HTML属性赋值，也可以使用CSS形式，但不接受普通的CSS属性！只能使用SVG元素的专有样式属性。

(2)SVG图形的属性不属于HTMLDOM标准，只能使用核心DOM方法操作其属性： rect.setAttribute('','')

(3)使用JS动态创建SVG元素，1)用svg.innerHTML ='' 2)用document.createElementNS('', '')，不能使用document.createElement()

(4)SVG元素的nodeName都是纯小写形式！与普通的HTML元素不同！

练习：

(1)在SVG画布上绘制一个矩形，从左移动到右，同时填充颜色还在不停的随机改变

(2)根据如下的JSON数据，绘制柱状统计图——有坑！！

[

{"label": "1月", "value": 350},

{"label": "2月", "value": 300},

{"label": "3月", "value": 450},

{"label": "4月", "value": 380}

]

5.使用SVG绘制圆形

<circle r="" cx="" cy=""></circle>

练习：

(1)创建30个大小随机、颜色随机、透明度随机的圆形

提示：使用createElementNS()创建元素,setAttribute()修改属性

(2)点击某个圆形，其慢慢变大、变淡...... 直至消失，从DOM树上删除该元素

(3)实现“蜻蜓点水”效果，点击svg画布的某处，即在此处生成一个圆形，立即变大、变淡...直至消失

6.使用SVG绘制椭圆

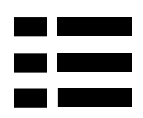
<ellipse rx="" ry="" cx="" cy=""></ellipse>

7.使用SVG绘制直线

<line x1="" y1="" x2="" y2="" stroke=""></line>

注意：所有的SVG图形默认只有填充色，没有描边色。

练习：使用line创建如下的图标：

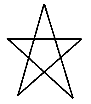


提示：可以把多个元素放在一个<g></g>小组中，可以自动继承小组的公共属性

8.使用SVG绘制折线

<polyline points="x1,y1 x2,y2 x3,y3 ....." stroke="" fill-opacity="0"></polyline>

练习：使用折线绘制如下的图标



9.使用SVG绘制多边形 17:25

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

练习：使用多边形绘制如下的图标：



10.使用SVG绘制文本

提示：传统的标签不能置于SVG内部！同理，SVG的标签也不能放在其它元素内部！

<text x="" y="" font-size="" alignment-baseline="before-edge">文本内容</text>

11 .使用SVG绘制图像

提示：在SVG中绘制图像使用image元素，引入位图后，此SVG图片放大后会失真。

<image width="" height="" xlink:href="x.jpg" x="" y=""></image>

12.如何使用渐变效果

<svg width="600" height="400" id="svg15">

<!--渐变属于特效，必须声明在“特效列表”-->

**<defs>**

<linearGradient id="g1" x1="0" y1="0" x2="100%" y2="0">

<stop offset="0" stop-color="#f00"></stop>

<stop offset="100%" stop-color="#0f0"></stop>

</linearGradient>

**</defs>**

<rect fill="url(#g1)"></rect>

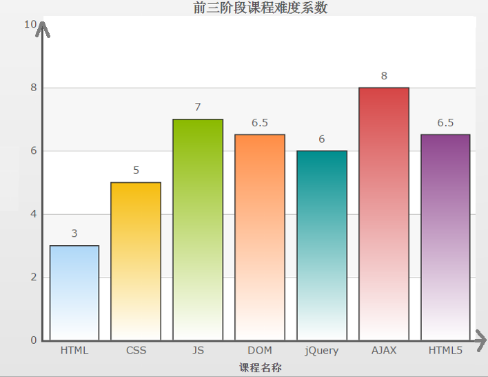
</svg>

课后练习：

1. 假设前端页面获取到后台返回了如下的JSON数据：

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

根据这段JSON字符串，绘制下图：



3)自学two.js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，月亮绕着地球转，地球绕着太阳转：

