复习：

H5新特性 —— 10个

(1)语义标签

(2)增强型表单

(3)视频和音频

(4)Canvas绘图

(5)SVG绘图

(6)地理定位

(7)拖放API

(8)Web Worker

(9)Web Storage

(10)WebSocket

Canvas绘图相关知识点

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

var ctx = $('#c').get(0).getContext('2d');

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

//绘制矩形

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

//绘制文本

ctx.strokeText()/fillText()/measureText().width

//绘制路径

ctx.beginPath()/closePath()

ctx.moveTo()/lineTo()

ctx.arc(cx, cy, r, start, end)

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

//绘制图像

var img = new Image()

img.src = "x.png"

img.onload =function(){

ctx.drawImage(img, x, y)

}

1.Canvas扩展小知识点：绘制变形后的图形/图像

CSS中有变形属性：transform： rotate/translate/scale/skew，这些变形属性只能用于特定的HTML DOM元素。

Canvas绘图只有一个元素：<canvas></canvas>，若向仅仅让其中的某个图形/图像变形，不能使用CSS变形属性！

**绘图上下文提供了专用的变形方法：**

ctx.rotate(旋转的弧度) 让画笔旋转指定的弧度

(1)画笔的旋转轴为画布的坐标原点

(2)画笔的旋转具有累加效果

(3)若想让绘图内容绕其它点旋转，必须平移画布的坐标原点

ctx.translate(x, y) 平移画布的坐标原点

(1)注意！画布原点平移，其上所有点的坐标值都发生改变！

ctx.scale()

ctx.save() 保存画笔当前的变形状态

ctx.restore() 恢复画笔变形状态到最近一次保存的值

练习： —— 非常抽象/比较难！！！

1.在画布的左上角绘制一个绕自己中心旋转的飞机；

1)平移画布原点到飞机1的中心;

2)旋转画笔

3)绘制飞机1

4)逆向旋转画笔

5)逆向平移画布的原点到左上角

2.在同一张画布右上角绘制一个不旋转的飞机；

6)绘制飞机2

3.在同一张画布左下角绘制绕自己中心旋转的飞机，速度是飞机1的2倍；

7)平移画布原点到飞机3的中心

8)旋转画笔

9)绘制飞机3

10)逆向旋转画笔

11)逆向平移画布的原点到左上角

2.作业题：使用Canvas绘制验证码图片

提示：此练习仅作为Canvas各种方法的练习！！

真正项目中验证码图片都是由服务器端(PHP、JSP、Node.js)技术来生成。

|  |
| --- |
| 总结：Canvas绘图技术  是H5新增的一种JS 2D 位图绘图技术，在项目中可用于统计图、小游戏、页面特效。 |

3.扩展：基于Canvas的第三方的统计图工具

(1)Chart.js 开源免费的图表函数库，8种图表

(2)FusionCharts 收费的图表工具库，90+种图表

(3)ECharts 百度提供的图表工具库

(4)FreeCharts 开源免费的图表函数库

.....

第三方工具库自学使用过程：

**(1)找到官网，查看说明**

http://www.chartjs.org/

Simple yet flexible JavaScript charting for designers & developers。

·Open source、

·8 Chart types、

·HTML5 Canvas、

·Responsive。

**(2)仿写官方的Demo程序**

var ctx = c10.getContext('2d');

new Chart(ctx, {

type: 'bar', //图表类型，共8种可用

data: { }, //绘图所用的数据

options: { } //图表选项，如禁用响应式、Y轴从0开始等

});

**(3)查看官方的API Document，在项目中加以应用**

练习：仿写Chart.js官方Demo，为柱状图添加不同的颜色

|  |
| --- |
| **位图：**bit图，每个图片都有若干个点组成，每个点由rgb描述，善于描述颜色的细腻变化，但放大会失真；适合于存储照片。  **矢量图**：每个图片由若干路径组成，每个路径由方向、大小、填充/描边色描述，可以无限放大而不会失真，但颜色变化不够细腻；适合于设计图形、Flash动画... |

4.H5新特性之五 —— SVG绘图

Scalable Vector Graph，可缩放的矢量图。

|  |  |  |
| --- | --- | --- |
|  | **Canvas绘图** | **SVG绘图** |
| 类型 | 位图 | 矢量图 |
| 放大 | 失真 | 不会失真 |
| 绘图类型 | JS绘图 | 元素/标签绘图 |
| 事件绑定 | 只能给整个画布绑定,非常不方便 | 很容易为某个图形/图像绑定监听 |
| Web项目中的适用场合 | 统计图、网页游戏 | 统计图、地图应用 |

SVG技术最早在2000年就有了，作为一种XML的扩展应用存在。后来H5标准把SVG标签纳入到H5内，进行了一部分精简。

**H5标准之前使用SVG的方法：**

编写一个XML文档，如my.svg，书写SVG标签；

再编写一个HTML文档，其中包含<img src="my.svg">

**H5标准之后使用SVG的方法：**

H5标签把SVG标签采纳进来，可以直接在H5文档中书写。

<svg>默认是一个300\*150的inline-block。注意：SVG画布的尺寸可以使用**CSS样式、HTML属性、JS对象**属性指定！SVG画布中不允许放置普通的HTML标签；SVG相关标签也不允许放在SVG画布外！

|  |
| --- |
| SVG图像/图形标签的共性：  (1)默认都没有尺寸，尺寸相关属性是必需！  (2)初始化都只有填充色(#000)，没有描边色  (3)SVG标签的样式可以用HTML标签属性形式声明，也可以类似于CSS形式声明；但SVG样式属性和CSS样式属性互不通用！  (4)可以使用核心DOM操作来访问SVG元素的属性(setAttribute/getAttribute)；不能使用HTMLDOM方式来操作。  (5)动态创建SVG图形标签可以使用拼接HTML字符串方式；也可以使用document.createElement**NS**('名称空间', '元素名') |

5.使用SVG的绘图标签

**(1)绘制矩形**

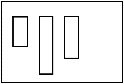
<rect **width="100" height="80"** x="0" y="320" fill="#f00" fill-opacity=".3" stroke="#a00" stroke-width="5" stroke-opacity=".8"></rect>

练习：

1)SVG画布正中央放置一个矩形，填充青色半透明，描边红色半透明；当鼠标移入该矩形，半透明都变为不透明；

2)SVG画布的左上角有一个矩形，点击一次，它就开始自动移动到右侧；

3)假设从服务器端得到如下数据：[ {label:'部门1',value:350},... ]，根据这些动态数据绘制统计图；



**(2)绘制圆形**

<circle **r="50"** cx="250" cy="200"></circle>

练习：随机的SVG画布上绘制30个圆形，大小随机、位置随机、填充颜色、填充透明度随机。点击某个圆时，它渐渐变大、变淡....直至消失，从DOM树上删除。

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

(3)绘制椭圆

(4)绘制直线

(5)绘制折线

(6)绘制多边形

(7)绘制文本

<text>文本内容</text>

(8)绘制图像

课后练习：

1. 使用Node.js+Express创建Web服务器，可以向客户端响应如下资源

静态资源：

/stat.html

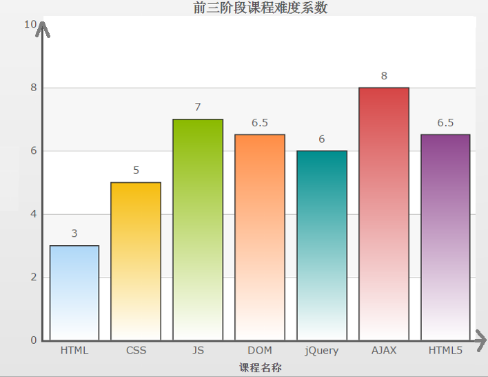
/js/jquery-1.11.3.js

动态资源：

GET /stat，该URL向客户端返回如下数据：

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

客户端请求得到/stat.html后，异步请求/stat，根据JSON数据绘制下图：



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：

