

SVG Learning Notes

简介

SVG: Scalable Vector Graphics 可缩放矢量图形

是基于 XML 来描述二维图形和绘图程序的语言

特点:

1. 基于 XML, 是纯粹的 XML
2. 在图像放大或改变尺寸时图形质量不会损失
3. 是 W3C 推荐标准
4. 文本编辑器读写
5. 和 JPEG 和 GIF 比起来, 尺寸更小, 可压缩性更强
6. 可伸缩
7. 任何分辨率下高质量打印
8. 图片质量不下降的情况下被放大
9. 文本是可选的, 也是可以搜索的, 适合制作地图
10. 可以和 java 技术一起运行
11. 是开放的标准
12. 主要竞争者: Flash
13. 和 Flash 的区别: SVG 是和其他标准相兼容(XSL\DOM), 而 flash 是未开源的私有技术
14. 几乎所有浏览器都支持

书写重点:

1. 元素和属性必须按标准,, 因为 XML 是区分大小写的 (这一点和 html 不同)
2. 属性值必须用引号引起来, 就算是数值也必须这样做。

编写流程:

1. 从 svg 根元素开始:
 - a) 应舍弃来自(X)HTML 的 doctype 声明, 因为基于 SVG 的 DTD 验证导致的问题比它能解决的问题更多。
 - b) 属性 **version** 和 **baseProfile** 属性是必不可少的, 供其它类型的验证方式确定 SVG 版本。
 - c) 作为 XML 的一种方言, SVG 必须正确的绑定命名空间 (在 **xmlns** 属性中绑定)。
2. 在 svg 标签中编写

实例:

```
<svg version="1.1"
      baseProfile="full"
      width="300" height="200"
      xmlns="http://www.w3.org/2000/svg">
```

```
<rect width="100%" height="100%" fill="red" />

<circle cx="150" cy="100" r="80" fill="green" />

<text x="150" y="125" font-size="60" text-anchor="middle" fill="white">SVG</text>
</svg>
```

渲染顺序：后来居上，越是后面的越可见

web 上的 svg 文件可以直接在浏览器上展示，或者通过以下几种方法嵌入到 HTML 文件中：

1. 如果 HTML 是 XHTML 并且声明类型为 `application/xhtml+xml`, 可以直接把 SVG 嵌入到 XML 源码中。
2. 通过 `object` 元素引用 SVG 文件：
`<object data="image.svg" type="image/svg+xml" />`
3. `iframe` 元素引用 SVG 文件：
`<iframe src="image.svg"></iframe>`
4. SVG 可以通过 JavaScript 动态创建并注入到 HTML DOM 中。 这样具有一个优点，可以对浏览器使用替代技术，在不能解析 SVG 的情况下，可以替换创建的内容。

文件类型

SVG 文件有两种形式：

1. 普通 SVG 文件：包含 SVG 标记的简单文本文件。
.svg 扩展名
2. 压缩 SVG：某些应用（比如地图应用等）中使用时，SVG 文件可能会很大，SVG 标准同样允许 gzip 压缩的 SVG 文件。推荐使用 “.svgz”（全部小写）作为此类文件扩展名

坐标定位

SVG 使用的坐标系统或者说网格系统，和 Canvas 用的差不多（所有计算机绘图都差不多）以页面的左上角为 $(0,0)$ 坐标点，坐标以像素为单位，x 轴正方向是向右，y 轴正方向是向下

在 SVG 文档中的 1 个像素对应输出设备（比如显示屏）上的 1 个像素(是可以改变的)

矩形

```
<rect x="60" y="10" rx="10" ry="10" width="30" height="30"/>
```

6 个属性:

1. **X**: 矩形左上角的 **x** 位置
2. **Y**: 矩形左上角的 **y** 位置
3. **Width**: 矩形的宽度
4. **Height**: 矩形的高度
5. **Rx**: 圆角的 **x** 方位的半径(这个角的圆滑程度)
6. **Ry**: 圆角的 **y** 方位的半径

椭圆

```
<ellipse cx="75" cy="75" rx="20" ry="5"/>
```

1. **Rx** 椭圆的 **x** 半径(长半轴)
2. **Ry** 椭圆的 **y** 半径(短半轴)
3. **Cx** 椭圆中心的 **x** 位置
4. **Cy** 椭圆中心的 **y** 位置

线条

Line 绘制直线。它取两个点的位置作为属性, 指定这条线的起点和终点位置

```
<line x1="10" x2="50" y1="110" y2="150"/>
```

折线

Polyline 是一组连接在一起的直线。因为它可以有很多的点, 折线的的所有点位置都放在一个 **points** 属性中:

```
<polyline points="60 110, 65 120, 70 115, 75 130, 80 125, 85 140, 90 135, 95 150, 100 145"/>
```

points

点集数列。每个数字用空白、逗号、终止命令符或者换行符分隔开。每个点必须包含 2 个数字, 一个是 **x** 坐标, 一个是 **y** 坐标。

多边形

polygon 和折线很像, 它们都是由连接一组点集的直线构成。不同的是, **polygon** 的路径

在最后一个点处**自动回到第一个点**。需要注意的是，矩形也是一种多边形，如果需要更多灵活性的话，你也可以用多边形创建一个矩形

```
<polygon points="50 160, 55 180, 70 180, 60 190, 65 205, 50 195, 35 205, 40 190, 30 180, 45 180"/>
```

路径

path 可能是 SVG 中最常见的形状。你可以用 path 元素绘制矩形(直角矩形或者圆角矩形)、圆形、椭圆、折线形、多边形，以及一些其他的形状，例如贝塞尔曲线、2 次曲线等曲线。

G

一组

<https://segmentfault.com/a/1190000012071386>

