

SVG

陈一帅

实务学堂

介绍

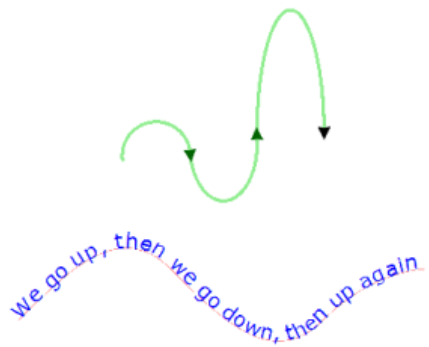
- 编程真的是什么都能干
- 本节我们学习如何通过写程序，画图，这就是SVG图
- 通过这种方法得到的图像，它的分辨率是无限大的，也就是说：不管你怎么放大，图片永远清晰。厉害吧

作品

- 先请大家欣赏一副作品
- 外星人来了

介绍

- Scalable Vector Graphics: 可缩放矢量图形
- 一种XML语言
- 文本, 曲线, 形状

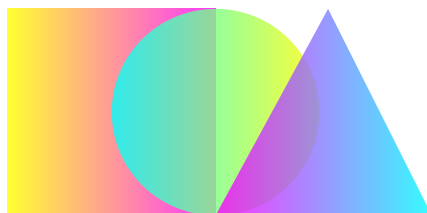


SVG

- 元素渲染顺序
 - SVG文件全局有效的规则是“后来居上”，越后面的元素越可见。
- 区分大小写
- 属性值必须用引号引起来，就算是数值也必须这样做。

可扩展

- 放大缩小，清晰度不变

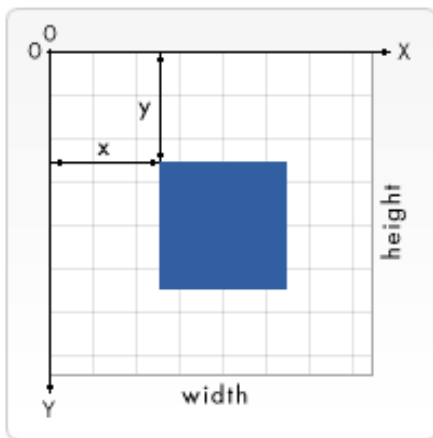


XML

- Extensible Markup Language
- 像 HTML

坐标系

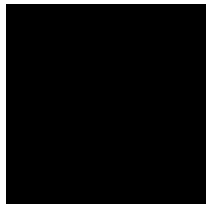
- 所有计算机绘图都差不多
 - 页面左上角为(0,0)坐标点
 - 坐标以像素为单位
 - x轴正方向是向右
 - y轴正方向是向下



SVG 画矩形

- 这和你小时候所教的绘图方式是相反的。但在HTML文档中，元素都用这种方式定位
- 定义一个矩形，从左上角开始，向右延展100px，向下延展100px，形成一个100*100大的矩形

```
<rect x="0" y="0" width="100" height="100" />
```



SVG 画矩形

```
<svg width="300" height="200">
```

```
  <!-- 绘制一个完全覆盖图像区域的矩形，背景设为红色 -->
```

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

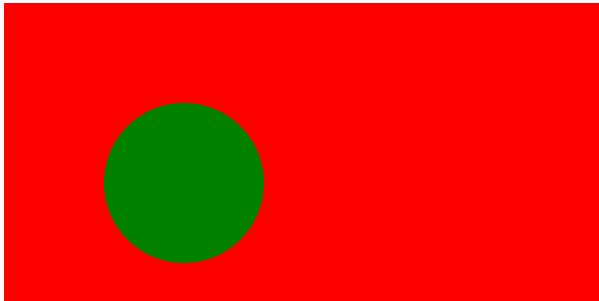
```
</svg>
```



圆卷

- 向右cx: 偏移90px; 向下cy: 偏移90px
- 半径r: 40px
- 填充fill: 绿色 green

```
<circle cx="90" cy="90" r="40" fill="green"/>
```



文字

- 填充 (fill) 为白色 (red)
- 文本 (text) 锚点 (anchor) 居中 (middle)
- 精细调整字体 (font) 大小 (size)
- 垂直位置 (x, y)

```
<text x="150" y="85" font-size="60"  
      text-anchor="middle" fill="red"> 实务学堂  
</text>
```



实务学堂

线

```
<line x1="10" x2="50" y1="110" y2="150"  
stroke="orange" fill="transparent"  
stroke-width="5"/>
```



折线

```
<polyline points="60 110 65 120 70 115 75  
130 80 125 85 140 90 135 95 150 100 145"  
stroke="orange" fill="transparent"  
stroke-width="5" />
```



椭圆

```
<ellipse cx="75" cy="75" rx="20" ry="5"  
stroke="red" fill="transparent"  
stroke-width="5"/>
```



MDN

路径

- 最常见的SVG元素
- 可用path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。
- 很强大，很复杂
- 只需要设定很少的点，就可以创建平滑流畅的线条（比如曲线）
 - polyline元素也能实现类似的效果，但必须设置大量的点（点越密集，越接近连续，看起来越平滑流畅），并且这种做法不能够放大（放大后，点的离散更明显）。所以在绘制SVG时，对路径的良好理解很重要。

Path

- 通过属性d定义
- 属性d的值是一个“命令+参数”的序列

命令+参数

- 每一个命令都用一个关键字母来表示
 - 字母“M”表示的是“Move to”命令
 - 当解析器读到这个命令时，它就知道你是打算移动到某个点
- 跟在命令字母后面的，是你需要移动到的那个点的x和y轴坐标
 - “M 10 10”：移动到(10,10)这个点
 - 这一段字符结束后，解析器就会去读下一段命令

命令

- 每一个命令都有两种表示方式
- 大写字母，表示采用绝对定位
- 小写字母，表示采用相对定位
 - 从上一个点开始，向上移动10px，向左移动7px
- 因为属性d采用的是用户坐标系，所以不需标明单位。

Path 命令

- M: move to 移动
- L: line to 连线
- H: horizontal line to 水平连线
- V: vertical line to 垂直连线
- C: curve to 曲线

[MDN](#)

移动命令 M

- M命令是移动画笔位置，但是不画线
- 两个参数，分别是需要移动到的点的x轴和y轴的坐标
- 常出现在路径的开始处，用来指明从何处开始画

画线： Line to

- 在当前位置和新位置（L前面画笔所在的点）之间画一条线
- L需要两个参数，分别是一个点的x轴和y轴坐标
- 两个简写命令，用来绘制水平线和垂直线
 - H，绘制水平线
 - V，绘制垂直线
 - 都只带一个参数，标明在x轴或y轴移动到的位置，因为它们都只在坐标轴的一个方向上移动。

例

- 矩形（同样的效果用元素可以更简单的实现）

```
<path d="M10 10 H 90 V 90 H 10 L 10 10" />
```



Z 闭合

- 可通过“闭合路径命令”Z来简化上面的path
- Z命令会从当前点画一条直线到路径的起点
- 尽管我们不总是需要闭合路径，但是它还是经常被放到路径的最后
- Z命令不用区分大小写。

```
<path d="M10 10 H 90 V 90 H 10 L 10 10" />
```

可以简化为：

```
<path d="M10 10 H 90 V 90 H 10 z" />
```

或者采用相对

```
<path d="M10 10 h 80 v 80 h -80 z" />
```


讨论：下面代码的图像是？

```
<path d="
    M 20 20
    H 180
    V 180
    H 20
    z">
</path>
```

讨论：下面代码的图像是？

```
<path d="
    M 100 0
    h 100
    v 100
    h 100
    v 100
    h -100
    v 100
    h -100
    v -100
    h -100
    v -100
    h 100
    z">
</path>
```

示例

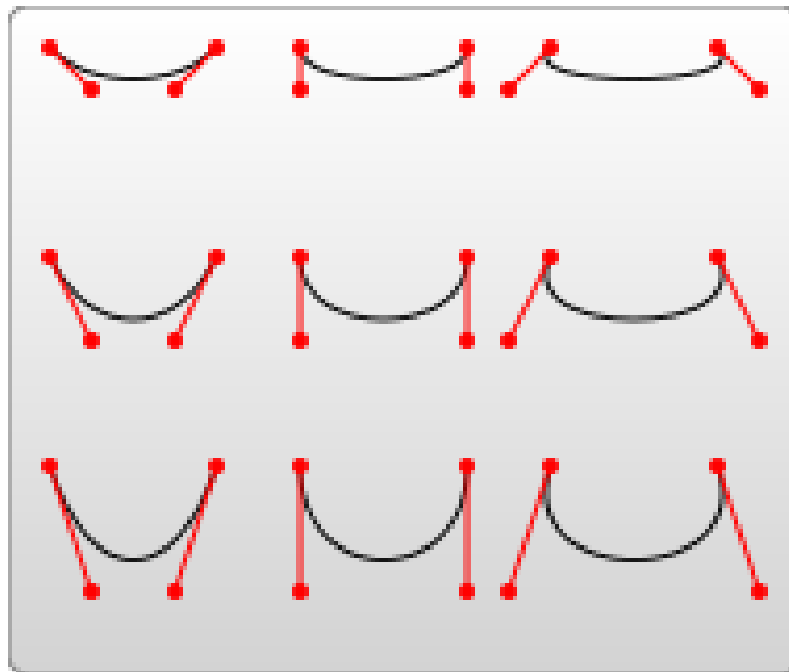
- 十字架

曲线

- 两种贝塞尔曲线
 - 三次贝塞尔曲线C
 - 二次贝塞尔曲线Q

三次贝塞尔曲线

- 需定义一个曲线终点和两个控制点
- 控制点描述的是曲线起始点的方向
- 曲线上各个点的方向，是从起点方向逐渐变到终点方向



三次贝塞尔曲线

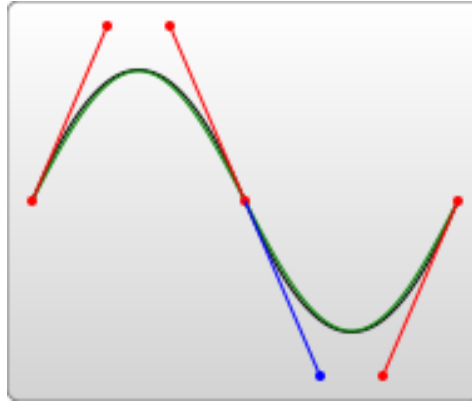
- C命令，三组坐标参数
 - C x1 y1, x2 y2, x y
 - c dx1 dy1, dx2 dy2, dx dy
- (x,y)表示曲线终点
- (x1,y1)是起点的控制点
- (x2,y2)是终点的控制点

曲线

```
<path d=" M 20 40  
  L 100 20  
  L 175 125  
  L 120 180  
  C 40 150 70 50 20 40">  
</path>
```

- 曲线

S 命令

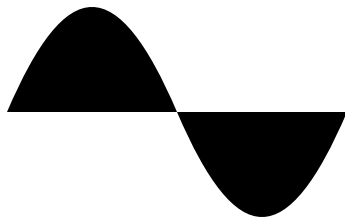


- 将若干个贝塞尔曲线连起来，创建出一条很长的平滑曲线
 - 一个点某一侧的控制点是它另一侧的控制点的对称（以保持斜率不变）
 - 如果S命令跟在一个C或S命令后面，则它的第一个控制点会被假设成前一个命令曲线的第二个控制点的中心对称点。
 - 如果S命令单独使用，前面没有C或S命令，那当前点将作为第一个控制点。

S命令

- 下面是S命令的语法示例，图中左侧红色标记的点对应的控制点即为蓝色标记点。

```
<path d="M10 80  
      C 40 10, 65 10, 95 80  
      S 150 150, 180 80"/>
```



Q

- 二次贝塞尔曲线Q
- 比三次贝塞尔曲线简单，只需要一个控制点，用来确定起点和终点的曲线斜率
- 因此它需要两组参数，控制点和终点坐标。

[MDN Path](#)

[MDN SVG](#)

工具

- Adobe Illustrator
- Inkscape
- Affinity
- [clker SVG 编辑器](#)

剪切

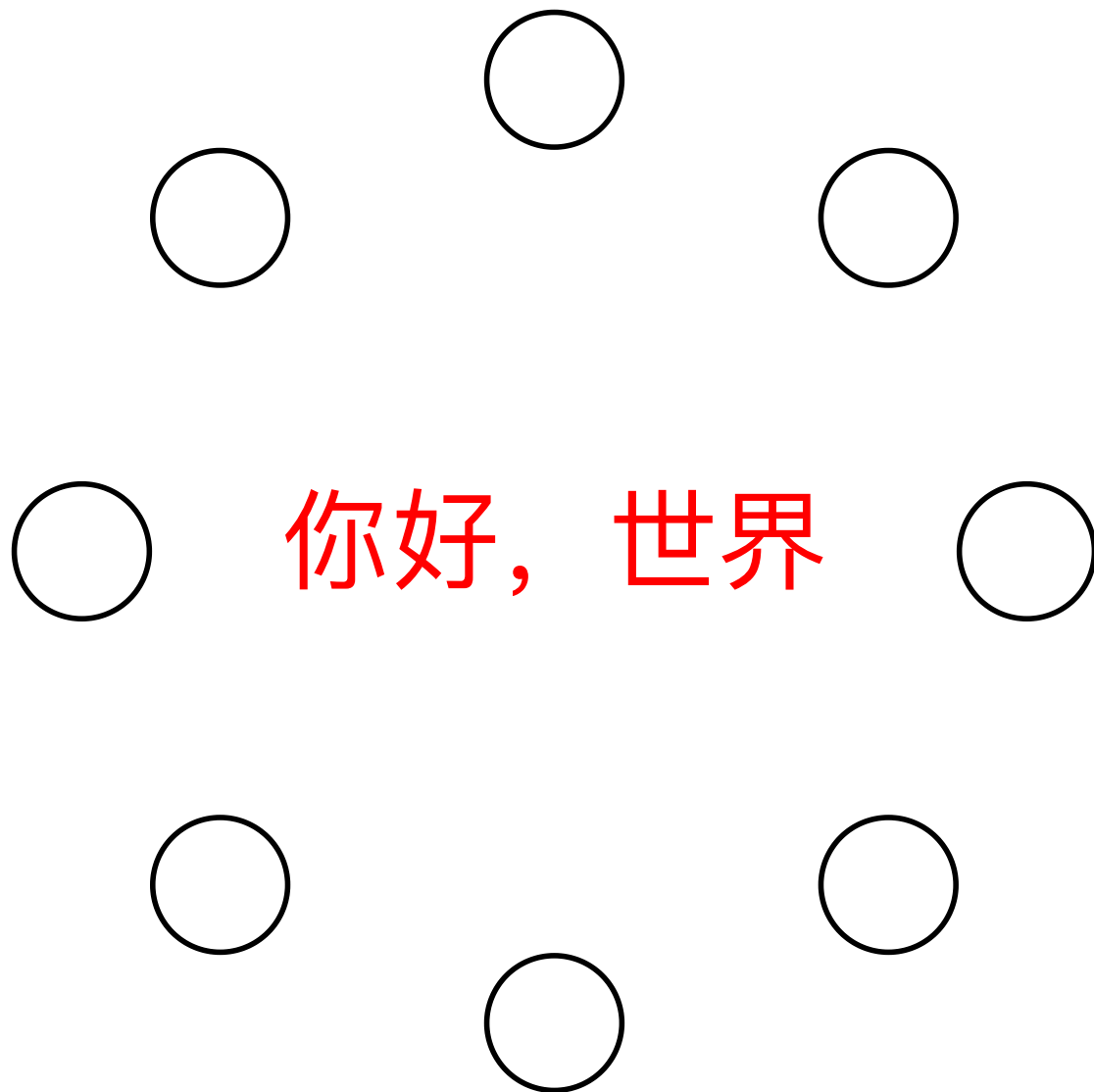
- clip-path

爱心

设置风格

- 类似CSS，设置图片风格
- 颜色过渡：lineargradient
 - Shape
 - Multi-Shape
- 自带滤镜：filter
 - Circle
 - Multi Circle

动画



练习

- 为学堂设计一个图标
- 参考：[icomoon](#)

练习

create an SVG icon system.

Vector Glyphs You can think of an icon system as a set of categorically-related symbols designed with a consistent aesthetic. Begin by considering what category or theme you would like to design your icon system around. Once you decide, you may want to take some time to sketch out a few preliminary symbols that will guide your approach to the rest of them. Your icon system for the SVG assignment should consist of at least five original symbols.

Using a text editor and, optionally, illustration software, create a series of vector graphic symbols—each in its own document—that will comprise your icon system. When applied to a web page, your icon system will function similar to a text font. As such, you need not worry about the color of

要求

An original icon system of at least five vector symbols that are categorically and aesthetically related (5 points) A single SVG document that contains the code for the full icon system (2 points) A web page that displays each symbol of the icon system together (1.5 points) CSS for the color and size of each symbol, displayed uniformly across the page. (1.5 points)