

1. SVG简介

什么是SVG?

SVG (Scalable Vector Graphics) 是一种基于XML的矢量图形格式。与位图不同, SVG使用数学描述来定义图形, 可以无损缩放任意大小。

主要优势

- 矢量图形, 无损缩放
- 文件体积小
- 可直接嵌入HTML
- 支持CSS样式和JavaScript

简单示例



```
<svg width="150" height="100">
  <circle cx="75" cy="50" r="30"
    stroke="black" stroke-width="2"
    fill="red" />
</svg>
```

2. 语法简介

基本语法

- SVG使用XML格式, 以`<svg>`根元素开始。 `width`, `height`表示画布尺寸, `viewBox`表示可视区域
- SVG元素使用标签来描述不同的图形, 例如 `circle`表示圆形, `rect`表示矩形等
- SVG元素可以包含属性, 用于描述图形的各种特性, 如位置、大小、颜色等。

SVG的坐标系

- 原点(0,0)在左上角
- X轴向右递增
- Y轴向下递增

坐标系示意图



```
<svg width="200" height="200"
  xmlns="http://www.w3.org/2000/svg" viewBox="0 0 200 200">
  <circle cx="100" cy="100" r="50" fill="blue" />
</svg>
```



3. 基本形状

SVG提供了多种基本形状: 矩形(rect)、圆形(circle)、椭圆(ellipse)、线条(line)、多边形(polygon)。

</> 基本形状示例

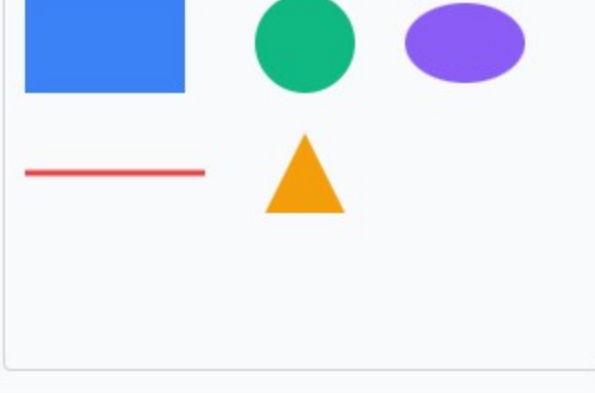
```
<svg width="300" height="200">
  <!-- 矩形 -->
  <rect x="10" y="10" width="80" height="50" fill="#3882f6" />

  <!-- 圆形 -->
  <circle cx="150" cy="35" r="25" fill="#108981" />

  <!-- 椭圆 -->
  <ellipse cx="230" cy="35" rx="35" ry="20" fill="#8b5cf6" />

  <!-- 线条 -->
  <line x1="10" y1="100" x2="100" y2="100"
    stroke="#ef4444" stroke-width="3" />

  <!-- 多边形 -->
  <polygon points="150,80 170,120 130,120" fill="#f59e00" />
</svg>
```



4. SVG路径

路径(path)是SVG中最强大的元素, 可以创建任何复杂的形状。路径使用一系列命令来定义图形的轮廓。

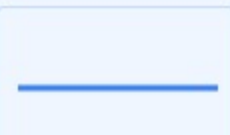
</> 基本语法

```
<path
  d="path-data"          <!-- 定义路径的路径数据 -->
  fill="fill-color"      <!-- 路径的填充颜色 -->
  stroke="stroke-color"  <!-- 路径的描边颜色 -->
  stroke-width="width"   <!-- 路径的描边高度 -->
/>

<!-- 路径命令 -->
M = moveto(移动)
L = lineto(画直线)
H = horizontal lineto(水平线到)
V = vertical lineto(垂直线到)
C = curveto(三次贝塞尔曲线到)
S = smooth curveto(光滑三次贝塞尔曲线到)
Q = quadratic Bézier curve(二次贝塞尔曲线到)
T = smooth quadratic Bézier curveto(光滑二次贝塞尔曲线到)
A = elliptical Arc(椭圆弧)
Z = closepath(关闭路径)
```

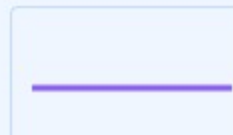
常用命令示例

直线路径 (L)



```
d="M 10 40 L 110 40"
```

水平线 (H)



```
d="M 10 40 H 110"
```

二次贝塞尔 (Q)



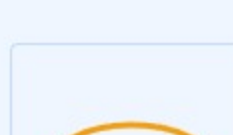
```
d="M 10 60 Q 60 20 110 60"
```

三次贝塞尔 (C)



```
d="M 10 60 C 30 100, 90 20, 110 60"
```

椭圆弧 (A)



```
d="M 20 60 A 40 20 0 0 1 100 60"
```

闭合路径 (Z)



```
d="M 30 20 L 90 20 L 60 60 Z"
```

5. SVG文本处理

SVG提供了强大的文本处理功能, 不仅可以显示普通文本, 还支持路径文本、多行文本、文本样式等高级特性。

T 基本文本元素

text/T元素语法

```
<text
  x="x-coordinate"      <!-- 文本左上角的 x 坐标 -->
  y="y-coordinate"      <!-- 文本左上角的 y 坐标 -->
  font-family="font"    <!-- 字体名称 -->
  font-size="size"      <!-- 字体大小 -->
  fill="fill-color"     <!-- 文本颜色 -->
  text-anchor="anchor"  <!-- 文本锚点 -->
>
  Text content          <!-- 文本内容 -->
</text>
```

重要属性:

x, y: 文本基线起始位置

text-anchor: 文本对齐方式

font-family: 字体族

font-size: 字体大小

font-weight: 字体粗细

基本文本示例

普通文本
粗体文本
斜体文本
下划线文本

5. 常用属性与特性

基本属性

```
<!-- 填充和描边 -->
fill="#000000"          <!-- 填充颜色 -->
stroke="#ff4444"         <!-- 描边颜色 -->
stroke-width="3"         <!-- 描边宽度 -->
opacity="0.7"           <!-- 透明度 -->
```



基本属性示例

渐变效果

```
<!-- 线性渐变 -->
<defs>
  <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
    <stop offset="0%" stop-color="#ff0000" />
    <stop offset="100%" stop-color="#4ecdc4" />
  </linearGradient>
</defs>
<rect fill="url(#grad1)" ... />

<!-- 径向渐变 -->
<linearGradient id="grad2" cx="50%" cy="50%" r="50%">
  <stop offset="0%" stop-color="#ff0000" />
  <stop offset="100%" stop-color="#000000" />
</linearGradient>
```



线性渐变和径向渐变

滤镜效果

```
<!-- 阴影效果 -->
<defs>
  <filter id="shadow">
    <feDropShadow dx="3" dy="3" stdDeviation="2" flood-color="#000" flood-opacity="0.5"/>
  </filter>
</defs>
<rect filter="url(#shadow)" ... />

<!-- 模糊效果 -->
<filter id="blur">
  <feGaussianBlur stdDeviation="3"/>
</filter>
```



阴影和模糊效果

5. SVG应用

SVG在现代Web开发中有着广泛的应用, 包括图标系统、数据可视化、动画效果等。本章将介绍SVG的主要应用场景和实际案例。

图标应用

SVG图标优势

- 矢量特性: 无限缩放不失真
- 文件小: 比位图文件更小
- 可定制: 可通过CSS修改颜色、大小
- 交互性: 支持悬停、点击等效果
- 无障碍: 支持屏幕阅读器

注意事项:

在实际项目中, 我们通常不会手写SVG图标代码, 而是使用专业的设计工具如 **Figma**、**Adobe Illustrator** 或 **Sketch** 来制作图标, 然后导出为SVG格式。

常用图标示例



动画应用

动画原理说明

SVG支持多种动画效果, 包括属性动画、路径动画、变换动画等。由于PDF不支持动态效果, 这里展示静态的动画关键帧。

常用动画类型:

- 属性动画: 颜色、透明度变化
- 变换动画: 旋转、缩放、平移
- 路径动画: 沿路径运动
- 形状动画: 形状变化

```
<!-- 旋转动画示例 -->
<rect width="50" height="50" fill="blue">
  <animateTransform
    attributeName="transform"
    type="rotate"
    values="0 25 360 25 25"
    dur="2s"
    repeatCount="indefinite"/>
</rect>
```

动画关键帧展示



数据可视化应用

简单柱状图



简单饼图

