

SVG 语法教程指南中文版

感谢：翻译蔡发明，YAO-2100 整理。

1 SVG 简介

SVG 是由 W3C 制定的基于可扩展标记语言 (XML) 来描述二维矢量图型的一个开放标准。

1.1 学习之前应具备的知识基础

你应该确保你有以下知识基础以便在后面更好的学习和理解 SVG，并达到事半功倍的效果：

- * HTML
- * Basic XML

如果你想先学习一下这些知识。请访问 W3Schools Online Web Tutorials (英文站点)，中文内容可以访问 SVG 中国的基础内容栏目。

1.2 什么是 svg?

- * SVG 指的是 Scalable Vector Graphics，“可升级矢量图形”的意思；
- * SVG 是用来定义面向 web 的矢量图形；
- * SVG 基于 XML 语言定义图形；
- * 当你放大或重新设置图形大小时，SVG 图形质量不会有所损失；
- * SVG 里的每一个元素和属性都可以自定义；
- * SVG 是一个 W3C 推荐标准；
- * SVG 可以和 W3C 的其他推荐标准结合使用。

1.3 SVG 的历史和优点

Web 浏览器在表现矢量图形方面的薄弱，引起各大软件厂商和组织纷纷推出自己的矢量图形规范，为统一标准结束这种混乱局势，W3C 组织于 1998 年 8 月专门成立了 SVG 工作组致力于图形标准的制定工作，同时 Microsoft、Sun、HP、IBM、Apple、Adobe、Corel 及 Kodak 等一些著名公司也参与到 SVG 标准的制定。

W3C 于 1999 年 2 月 11 日发布了 SVG 第一个讨论草案；

W3C 于 2001 年 9 月 4 日发布 SVG 1.0；

W3C 于 2003 年 1 月 4 日发布 SVG 1.1；

W3C 于 2003 年 1 月 14 日推出 SVG 移动子版本：SVG Tiny 和 SVG Basic；

W3C 于 2005 年 4 月日发布的 SVG 1.2 草案；

W3C 于 2006 年 8 月 10 日 SVG Tiny 1.2 为 W3C 候选推荐标准。

与其他图形格式 (JPEG、GIF) 相比使用 SVG 的优点如下：

- * SVG 文件可以被很多的工具（比如记事本）阅读和编辑；
- * SVG 文件比 JPEG 和 GIF 更小压缩比更大；
- * SVG 图像是可压缩的；
- * SVG 图像可以以任何分辨率高清晰打印；
- * SVG 图像是可缩放的，图像的任何部分都可以无损失缩放；
- * SVG 里的文字是可供选择和搜索（制作地图的良好功能）；
- * SVG 可以和诸如 java 等技术共同工作；
- * SVG 是一个开放标准；
- * SVG 文件是纯 xml 的。

SVG 的主要竞争对手是 Flash。

SVG 与 Flash 两者有很多的相似特征，但和 Flash 相比 SVG 最大优点是他与其他标准（比如 XSL 和 DOM）相兼容。而 Flash 则是未开源的私有专利技术，只属 Adobe 公司所有，不具有开放性。

1.4 查看 SVG 文件

SVG 的最大缺点是没有完全支持 SVG 的浏览器，准确的说是主流浏览器不支持 SVG 文件的显示。不过当前越来越多的厂商已经投入到开发 SVG 项目中，并且很多著名企业已经发布支持 SVG 的程序软件，SVG 阵营正在迅猛扩大。

Opera 浏览器支持 SVG 显示。

Mozilla Firefox 浏览器自版本 1.5 发行后，开始支持 SVG 显示。

加载了 Adobe SVG Viewer 的 Internet Explorer 也能支持多数 SVG 特性，但是由于没有浏览器的原生支持，在交互方面许多特性不支持。不过 Microsoft 正计划 Internet Explorer 8.0 内置支持 SVG 显示。

我们等待 SVG 在更多流行的浏览器软件上支持，到那时 SVG 的时代就将带来。

二、SVG 实例

SVG 是用 XML 来编写的。

2.1 SVG 的一个实例

下面的例子是一个 SVG 的简单实例，SVG 文件必须保存扩展名为 “.svg” 格式。

```
<?xml version="1.0" standalone="no"?>
```

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
```

```
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
```

```
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
```

```
<circle cx="100" cy="50" r="40" stroke="black"
stroke-width="2" fill="red"/>

</svg>
```

查看例子（仅用于支持 SVG 的浏览器）

（如需查看 SVG 源代码，请打开此例，然后在窗口中右击。选择“查看源代码”。）

2.2 代码解释

第一行包含了 XML 声明。请注意 standalone 属性，这个属性规定本 SVG 文件是独立的还是包含一个对外部文件的引用。

standalone="no"意味着本 SVG 文档包含一个对外部文件的引用，比如说 DTD 文件。

第二和第三行引用了这个外部的 SVG DTD。该 DTD 位于“<http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd>”。该 DTD 位于 W3C，含有所有允许的 SVG 元素。

SVG 文档以根元素<svg>开始，包含开始标签<svg>和结束标签</svg>。width 和 height 属性用来设置 SVG 文档的宽和高，Version 属性定义了 SVG 的版本，xmlns 属性定义了 SVG 的名称空间。

<circle>元素用来划一个圆。cx 和 cy 属性定义了圆中心点的 x，y 坐标。如果 cx，cy 省略，则圆以（0，0）点为中心。r 属性定义了圆的半径。

Stroke 和 stroke-width 属性控制图形的外轮廓。我们给圆设置了一个宽 2px 的黑色边框。

fill 属性指出了图形的颜色。我们给圆设置了红色。

</svg>标签则结束了 svg 根元素和本文档。

注意：所有开始标签必须有对应的结束标签。

三、在 HTML 页面中的插入 SVG

SVG 文件可通过以下标签嵌入 HTML 文档：<embed>、<object>或者<iframe>。

下面你将看到 SVG 嵌入到 HTML 文档的三种方法。

3.1 使用<embed>标签

所有主流浏览器都支持<embed>标签，允许使用脚本。

注释：Adobe SVG Viewer 推荐您使用<embed>标签来将 SVG 包含进 HTML 文档！但是，当你要创建有效的 XHTML 文档的时候，你不能使用<embed>。因为任何一个 HTML 的推荐标准里都不推荐<embed>标签。

语法：

```
<embed src="/rect.svg" width="300" height="100"
```

```
type="image/svg+xml"
pluginspage="http://www.adobe.com/svg/viewer/install/" />
```

注释: `pluginspage` 属性指出了插件下载的地址。

提示: IE 浏览器支持一个附加的属性, `wmode="transparent"`, 用来使 HTML 背景透明。

3.2 使用<object>标签

<object>标签是 HTML 4 的标准标签, 被所有新的浏览器支持。缺点是他不支持脚本。

注意: 如果你安装的最近版本的 Adobe SVG Viewer, 运用<object>标签 SVG 文件将不工作 (至少不能在 IE 中工作)。

语法:

```
<object data="rect.svg" width="300" height="100"
```

```
type="image/svg+xml"
codebase="http://www.adobe.com/svg/viewer/install/" />
```

注释: `codebase` 属性指出插件下载的地址。

3.3 使用<iframe>标签

<iframe>标签可以在大部分的浏览器里工作。

语法:

```
<iframe src="/rect.svg" width="300" height="100">
</iframe>
```

3.4 我最期望的方式

如果我们可以运用 SVG 命名空间, 直接将 SVG 元素添加到 HTML 里, 那真是太棒了! 就像下面这个样子:

```
<html
xmlns:svg="http://www.w3.org/2000/svg">
<body>
```

```
<p>This is an HTML paragraph</p>
```

```
<svg:svg width="300" height="100" version="1.1" >
<svg:circle cx="100" cy="50" r="40" stroke="black"
```

```
stroke-width="2" fill="red" />
</svg:svg>

</body>
</html>
```

四、SVG<rect>

Svg 有一些预定义的可以直接使用操作的图形元素

```
Rectangle <rect>
Circle <circle>
Ellipse <ellipse>
Line <line>
Polyline <polyline>
Polygon <polygon>
Path <path>
```

下面将详细介绍每一个元素，从 rect 开始。

The <rect> Tag

<rect>用来生成矩形或矩形的变种。

为了理解<rect>怎么工作，复制下面的代码到记事本，然后保存为 rect1.svg 到你的网页目录。

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<rect width="300" height="100"
style="fill:rgb(0,0,255);stroke-width:1;
stroke:rgb(0,0,0)"/>
</svg>
```

代码解释：

Width 和 height 属性指出矩形的宽和高。

style 属性定义 css 属性

fill 属性定义矩形填充的颜色（rgb 值，颜色值，十六进制值）

stroke-width 属性指定矩形填充线宽。

stroke 属性指定矩形 border 颜色。

再看一个带有新属性的例子

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<rect x="20" y="20" width="250" height="250"
style="fill:blue;stroke:pink;stroke-width:5;
fill-opacity:0.1;stroke-opacity:0.9"/>
</svg>
```

代码解释:

x 属性定义矩形的左位置 (例如 x="0"将矩形放在距 window 窗口左端 0px 处)

y 属性定义矩形距上端的位置 (例如 y="0"将矩形放在距 window 窗口上部 0px 处)

CSS fill-opacity 属性定义填充颜色的透明度 (从 0 到 1)

CSS stroke-opacity 属性定义描边颜色的透明度 (从 0 到 1)

为整个元素定义元素例子

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<rect x="20" y="20" width="250" height="250"
style="fill:blue;stroke:pink;stroke-width:5;
opacity:0.9"/>
</svg>
```

代码解释:

CSS opacity 属性定义整个元素的透明度 (从 0 到 1)

下一个例子, 创建一个圆角矩形

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<rect x="20" y="20" rx="20" ry="20" width="250"
height="100" style="fill:red;stroke:black;
stroke-width:5;opacity:0.5"/>
```

</svg>

代码解释:

rx and ry 属性定义举行的圆角

五、SVG <circle>

<circle>属性用来创建一个圆。

将下列代码复制进记事本，并保存成.svg，放在你的 web 目录。

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<circle cx="100" cy="50" r="40" stroke="black"
stroke-width="2" fill="red"/>
</svg>
```

[View example](#)

代码解释:

cx and cy 属性定义圆心的 x, y 坐标，如果省略，在圆心为 (0, 0)

r 属性定义圆的半径

六、SVG<ellipse>

<ellipse>标签用来创建一个椭圆。椭圆类似于圆. 不同仅仅在于椭圆有 x, y 两个值不同半径，而圆的 x, y 半径值相同。

复制下面代码到记事本，存为 ellipse1.svg，放在你的 web 目录里。

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<ellipse cx="300" cy="150" rx="200" ry="80"
style="fill:rgb(200,100,50);
stroke:rgb(0,0,100);stroke-width:2"/>
</svg>
```

代码解释:

cx 指出椭圆的中心点的 x 坐标

cy 指出椭圆的中心点的 y 坐标

rx 指出椭圆的横向半径

ry 指出椭圆的纵向半径

下面例子创建三个叠加的椭圆：

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<ellipse cx="240" cy="100" rx="220" ry="30"
style="fill:purple"/>
<ellipse cx="220" cy="70" rx="190" ry="20"
style="fill:lime"/>
<ellipse cx="210" cy="45" rx="170" ry="15"
style="fill:yellow"/>
</svg>
View example
```

下面例子将两个椭圆组合：

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<ellipse cx="240" cy="100" rx="220" ry="30"
style="fill:yellow"/>
<ellipse cx="220" cy="100" rx="190" ry="20"
style="fill:white"/>
</svg>
View example
```

七、SVG <line>

<line>标签用来创建一条直线。

复制下列代码到记事本，存成 line1.svg, 放在你的 web 目录下

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<line x1="0" y1="0" x2="300" y2="300"
style="stroke:rgb(99,99,99);stroke-width:2"/>
```


</svg>

代码解释:

x1 指定起点的 x 坐标

y1 指定起点的 y 坐标

x2 指定终点的 x 坐标

y2 指定终点的 y 坐标

八、SVG<polygon>

<polygon>标签用来创建一个多于三个边的图形。

父子下面代码倒记事本，存成 polygon1.svg, 放在你的 web 目录下

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<polygon points="220,100 300,210 170,250"
style="fill:#cccccc;
stroke:#000000;stroke-width:1"/>
</svg>
```

代码解释:

Points 属性定义多边形每个角的 x, y 坐标

下面例子创建一个四边形

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<polygon points="220,100 300,210 170,250 123,234"
style="fill:#cccccc;
stroke:#000000;stroke-width:1"/>
</svg>
```

九、SVG <polyline>

<polyline>用来创建由任意直线组成的图形

复制下列代码到记事本，存成 polyline1.svg, 放在你的 web 目录下

```
<?xml version="1.0" standalone="no"?>
```

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<polyline points="0,0 0,20 20,20 20,40 40,40 40,60"
style="fill:white;stroke:red;stroke-width:2"/>
</svg>
```

十、SVG<path>

<path>标签用来创建一条路径

M = moveto

L = lineto

H = horizontal lineto

V = vertical lineto

C = curveto

S = smooth curveto

Q = quadratic Belzier curve

T = smooth quadratic Belzier curveto

A = elliptical Arc

Z = closepath

注意：上面所有的命令也可以表示成小写形式。大写字母表示绝对位置，小写字母表示相对位置。

复制下面代码到记事本，存成 path1.svg, 放到你的 web 目录下。

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<path d="M250 150 L150 350 L350 350 Z" />
</svg>
```

上面的代码定义了一条起始自（250，150）直线到（150，350）接着直线到（350，350）最后在（250，150）闭合结束。

下面的例子创建了一条落选。

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
```

```
<path d="M153 334
C153 334 151 334 151 334
C151 339 153 344 156 344
C164 344 171 339 171 334
C171 322 164 314 156 314
C142 314 131 322 131 334
C131 350 142 364 156 364
C175 364 191 350 191 334
C191 311 175 294 156 294
C131 294 111 311 111 334
C111 361 131 384 156 384
C186 384 211 361 211 334
C211 300 186 274 156 274"
style="fill:white;stroke:red;stroke-width:2"/>
</svg>
```

复杂?是的!由于 path 的复杂性, 强烈建议使用 svg 编辑器来创建复杂的图形。

十一、SVG 滤镜(SVG Filters)

svg 滤镜用来给图形和文字添加特殊效果
可用的滤镜有以下这些:

- * feBlend
- * feColorMatrix
- * feComponentTransfer
- * feComposite
- * feConvolveMatrix
- * feDiffuseLighting
- * feDisplacementMap
- * feFlood
- * feGaussianBlur
- * feImage
- * feMerge
- * feMorphology
- * feOffset
- * feSpecularLighting
- * feTile
- * feTurbulence
- * feDistantLight
- * fePointLight
- * feSpotLight

提示: 你可以在每个元素里多重运用滤镜。

SVG 高新模糊 (Blur 值)

一个 svg 滤镜必须在<defs>标签里定义。

高斯模糊

<filter>标签用来定义一个 svg 滤镜。<filter>标签有一个必须的 id 属性来唯一标识该滤镜应用到图像中。

<filter>标签必须嵌套到<defs>标签里，<defs>标签是 definitions 的简写，允许特殊标签的定义。如 filter

复制下面代码到记事本，存为 filter1.svg，放到你的 web 目录下

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<defs>
<filter id="Gaussian_Blur">
<feGaussianBlur in="SourceGraphic" stdDeviation="3" />
</filter>
</defs>
<ellipse cx="200" cy="150" rx="70" ry="40"
style="fill:#ff0000;stroke:#000000;
stroke-width:2;filter:url(#Gaussian_Blur)"/>
</svg>
```

代码解释：

<filter>的 id 属性定义 filter 的唯一标识名(滤镜可以文档中的多个元素使用)

filter:url(#Gaussian_Blur)属性用来给元素链接滤镜。当链接到 filters 的 id 属性时，#是必须的。

滤镜效果是在<feGaussianBlur>标签里定义的。fe 前缀对所有滤镜有用

<feGaussianBlur>stdDeviation 属性用来定义 blur 的值

in="SourceGraphic"定义本效果的输入源是整个图像

不同 blur 值得另一个例子

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<defs>
<filter id="Gaussian_Blur">
```

```

<feGaussianBlur in="SourceGraphic" stdDeviation="20"/>
</filter>
</defs>
<ellipse cx="200" cy="150" rx="70" ry="40"
style="fill:#ff0000;stroke:#000000;
stroke-width:2;filter:url(#Gaussian_Blur)"/>
</svg>

```

十二、SVG 渐变 (SVG Gradients)

一个 svg 渐变必须在<defs>标签里定义

svg 渐变

渐变是自一种颜色到另一种颜色的逐渐过渡。另外，多个颜色渐变可以应用到相同的元素上。

有两种形式的渐变：

- * Linear Gradients
- * Radial Gradients

线性渐变 (Linear Gradients)

<linearGradient>标签用来定义一个 svg 线性渐变

<linearGradient>标签必须嵌套在<defs>标签里。<defs>标签是 definitions 的简写，允许特殊标签的定义，比如 gradients。

线性渐变可以被定义为，水平，垂直，角形渐变。

当 $y_1 \neq y_2$, $x_1 = x_2$ 时，产生水平渐变

当 $x_1 \neq x_2$, $y_1 = y_2$ 时，产生垂直渐变

当 $y_1 \neq y_2$, $x_1 \neq x_2$ 时，产生角形渐变

复制下列代码到记事本，保存为 linear1.svg, 放在你的 web 目录下。

```

<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<defs>
<linearGradient id="orange_red" x1="0%" y1="0%" x2="100%" y2="0%">
<stop offset="0%" style="stop-color:rgb(255,255,0);
stop-opacity:1"/>
<stop offset="100%" style="stop-color:rgb(255,0,0);
stop-opacity:1"/>
</linearGradient>
</defs>
<ellipse cx="200" cy="190" rx="85" ry="55"

```

```
style="fill:url(#orange_red)"/>
</svg>
```

代码解释:

<linearGradient>标签的 id 属性定义渐变的唯一标识名。
fill:url(#orange_red) 属性给椭圆链接渐变
<linearGradient>标签的 x1, x2, y1, y2 属性用来定义渐变的首尾位置
渐变颜色范围可以由两种或多种颜色组成。每一中颜色由一个<stop>标签描述。Offset 属性用来定义渐变首尾的颜色。

另一个例子

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<defs>
<linearGradient id="orange_red" x1="0%" y1="0%" x2="0%" y2="100%">
<stop offset="0%" style="stop-color:rgb(255,255,0);
stop-opacity:1"/>
<stop offset="100%" style="stop-color:rgb(255,0,0);
stop-opacity:1"/>
</linearGradient>
</defs>
<ellipse cx="200" cy="190" rx="85" ry="55"
style="fill:url(#orange_red)"/>
</svg>
```

径向渐变(Radial Gradients)

<radialGradient>标签用来定义一个 svg 径向渐变。
<radialGradient>标签必须嵌套在<defs>标签里。<defs>标签是 definitions 的简写, 允许特殊标签的定义, 比如 gradients。

复制下列代码到记事本, 保存为 linear1.svg, 放在你的 web 目录下。

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<defs>
<radialGradient id="grey_blue" cx="50%" cy="50%" r="50%"
```

```

fx="50%" fy="50%">
<stop offset="0%" style="stop-color:rgb(200,200,200);
stop-opacity:0"/>
<stop offset="100%" style="stop-color:rgb(0,0,255);
stop-opacity:1"/>
</radialGradient>
</defs>
<ellipse cx="230" cy="200" rx="110" ry="100"
style="fill:url(#grey_blue)"/>
</svg>

```

代码解释:

<radialGradient>的 id 属性定义了渐变的唯一标志名
fill:url(#gray_blue) 属性给椭圆链接渐变
cx,cy,r 属性定义外圆, fx,fy 定义内圆
渐变颜色范围可以由两种或多种颜色组成。每一中颜色由一个<stop>标签描述。Offset 属性用来定义渐变首尾的颜色。

另一个例子

```

<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<defs>
<radialGradient id="grey_blue" cx="20%" cy="40%" r="50%"
fx="50%" fy="50%">
<stop offset="0%" style="stop-color:rgb(200,200,200);
stop-opacity:0"/>
<stop offset="100%" style="stop-color:rgb(0,0,255);
stop-opacity:1"/>
</radialGradient>
</defs>
<ellipse cx="230" cy="200" rx="110" ry="100"
style="fill:url(#grey_blue)"/>
</svg>

```

十三、SVG 的元素(SVG Examples)

SVG Shapes

A rectangle

A rectangle with opacity

A rectangle with opacity 2

A rectangle with rounded corners
A circle
An ellipse
Three ellipses on top of each other
Two ellipses
A line
A polygon with three sides
A polygon with four sides
A polyline
A path
A spiral

SVG Filters

A feGaussianBlur filter
Another feGaussianBlur filter
A feBlend filter
A feColorMatrix filter
A feComponentTransfer filter
A feOffset filter
feOffset, feFlood, feComposite, feMerge, and feMergeNode
A feMorphology filter
* Filter 1
* Filter 2
* Filter 3
* Filter 4
* Filter 5
* Filter 6

SVG Gradients

Grad linear (horizontal gradient)
Grad linear (vertical gradient)
Grad radial
Grad radial 2

SVG Misc

Add an <a> element to a rectangle. The rectangle will act as a hyperlink
Use a JavaScript to create an a element when the user clicks on a circle
Rectangle that repeatedly fade away over 5 seconds
Animate the x, y, width, and height attributes of a rectangle and change the color of the rectangle
Three rectangles that will change color
Text that will move along a motion path
Text that will move, rotate and scale along a motion path

Text that will move, rotate and scale along a motion path + a rectangle that will "grow" and change color

十四、SVG Elements

The links in the "Element" column point to attributes and more useful information about the specific element.

[***表格]