Scalable Vector Graphics (SVG)

Scalable Vector Graphics (SVG) are part of the vector-based family of graphics. They are different from raster-based graphics, which store the color definition for each pixel in an array of data. The most common raster-based formats used on the web today are JPEG, GIF, and PNG, and each of these formats has strengths and weaknesses.

SVG has several advantages over any raster-based format:

- SVG graphics are created using mathematical formulas that require far less data to be stored in the source file because you don't have to store the data for each individual pixel.
- Vector images scale better. With images on the web, trying to scale an image up from its original size can result in distorted (or pixilated) images.

  The original pixel data was designed for a specific size. When the image is no longer that size, the program displaying the image guesses as to what data to use to fill in the new pixels. Vector images are more resilient; when the size of the image changes, the mathematical formulas can be adjusted accordingly.
- SVG images are rendered by the browser and can be drawn programmatically. They can be changed dynamically, making them especially suited for data-driven applications, such as charts.
- The source file for an SVG image is a text-based file, so it's both accessible and search engine friendly.

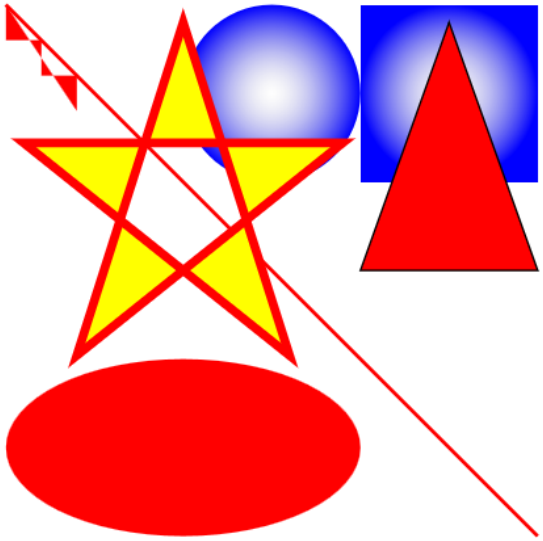
HTML5 allows us to embed SVG into HTML page directly.

Example 1

```
<svg width="300,300">
  <defs>
    <radialGradient id="gradient" cx="50%" cy="50%" r="50%" fx="50%" fy="50%">
      <stop offset="0%" stop-color="rgb(200,200,200)" stop-opacity="0" />
      <stop offset="100%" style="stop-color:rgb(0,0,255);stop-opacity:1" />
    </radialGradient>
  </defs>
  <rect id="redrect" x="200" width="100" height="100" style="fill:url(#gradient)" />
  <circle id="redcircle" cx="150" cy="50" r="50" fill="url(#gradient)" />
  <line x1="0" y1="0" x2="300" y2="300" style="stroke:red;stroke-width:2" />
  <ellipse cx="100" cy="250" rx="100" ry="50" fill="red" />
  <polyline points="0,0 0,20 20,20 20,40 40,40 40,60" fill="red" />

  <polygon points="250,10 300,150 200,150"
       style="fill: red; stroke: black; stroke-width: 1; border: dotted; fill-rule: evenodd" />
  <polygon points="100,10 40,198 190,78 10,78 160,198"
       style="fill:yellow;stroke:red;stroke-width:5;fill-rule:evenodd;" />
```

```
</svg>
```



Differences between Canvas and SVG:

Canvas	SVG
1. Canvas draws 2D graphics, on the fly (with a JavaScript). 2. Canvas is Resolution dependent 3. We don't have events in canvas. 4. Poor text rendering capabilities 5. You can save the resulting image as .png or .jpg from browser. 6. Renders fast compared to SVG	1. SVG is a language for describing 2D graphics in XML. 2. SVG is Resolution independent 3. SVG is XML based it supports event handlers (DOM) 4. Best suited for applications with large rendering areas (Google Maps) 5. SVC cannot be saved as image. 6. Slow rendering if SVG is complex.