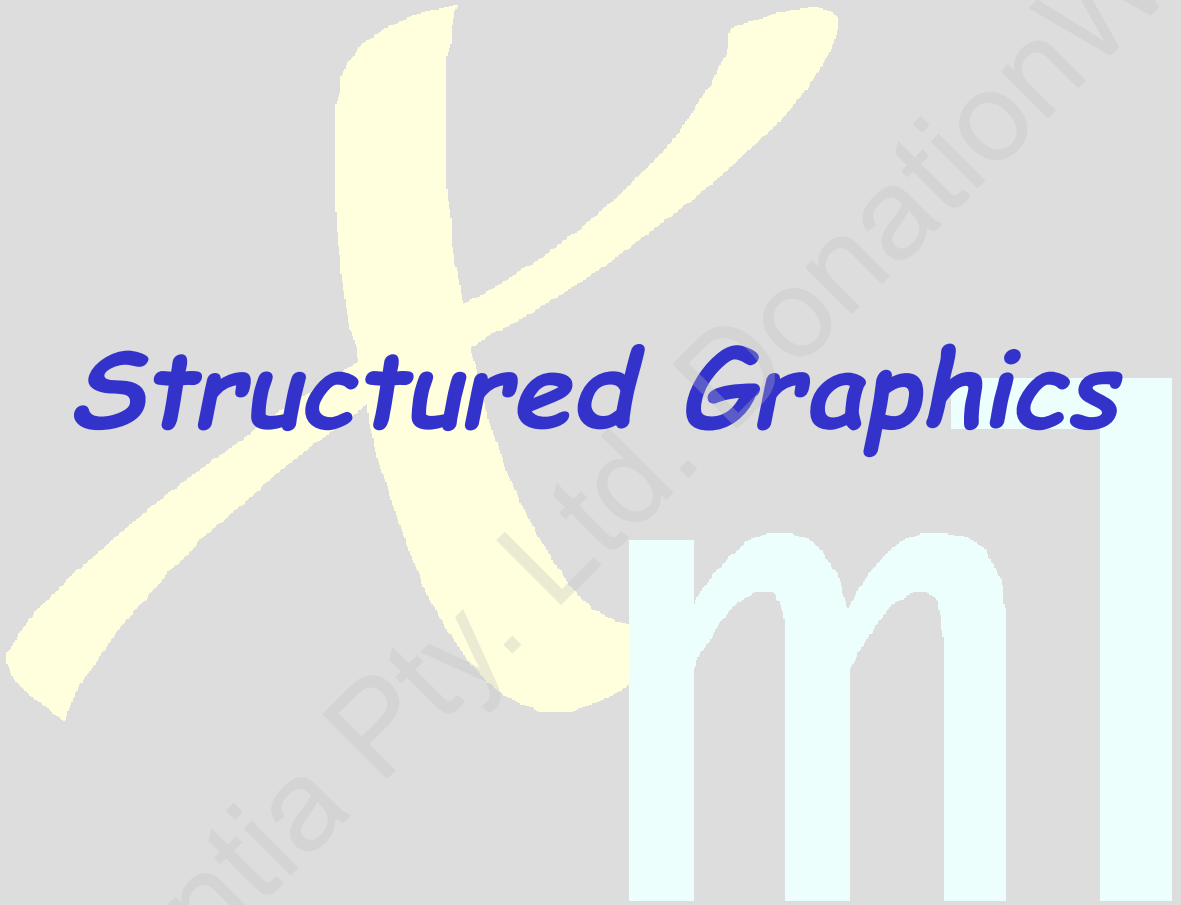
Structured Graphics



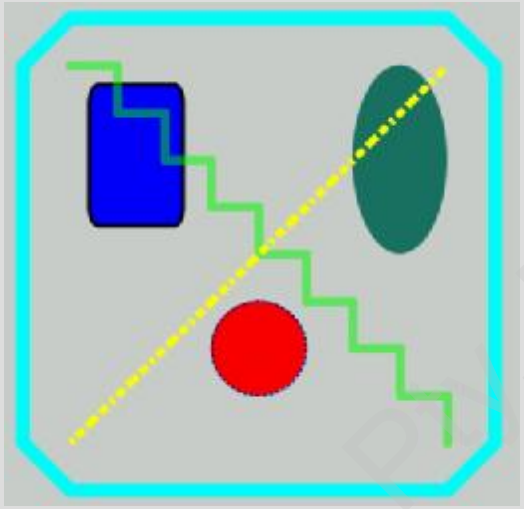
Scalable Vector Graphics

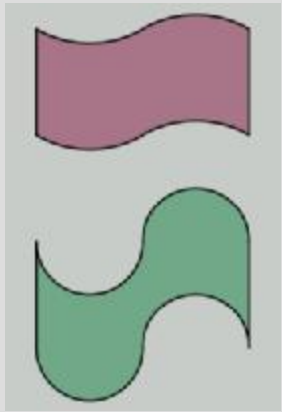
- **A language for describing two-dimensional graphics in XML**
  - raster (bitmap) graphics just don't work any more
    - *too many variations*
    - *hard to integrate into a system*
      - clickable maps are horrid
        - *hard to maintain*
      - can't be manipulated by script
        - *don't scale/transform, etc.*
    - *need a vector-based scheme*
  - currently being developed as a standard for web-based display of vector data such lines and polygons as well as images and text
    - *graphical objects can be grouped, styled, transformed and composited into previously rendered objects*
    - *feature set includes nested transformations, clipping paths, alpha masks, filter effects, template objects and extensibility*

Simple Example

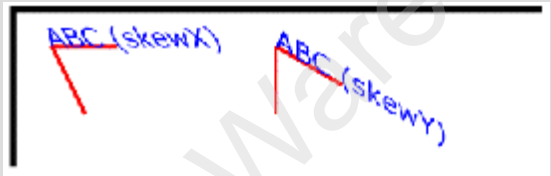
```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG December 1999//EN"
  "http://www.w3.org/Graphics/SVG/SVG-19991203.dtd">
<svg width="12cm" height="4cm" viewBox="0 0 1200 400">
  <desc>Example rect02 - rounded rectangles expressed in user coordinates</desc>

  <rect x="100" y="100" width="400" height="200" rx="50"
        style="fill:green;" />

  <g transform="translate(700 300); rotate(-30)">
    <rect x="0" y="0" width="400" height="200" rx="50"
          style="fill:none; stroke:purple; stroke-width:30" />
  </g>
</svg>
```

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg SYSTEM "svg-19990812.dtd">
<svg width="600" height="600">
  <title> Simple shapes </title>
  <rect x="120" y="120" width="100" height="150" rx="10" ry="20" style="fill:blue; stroke:black; stroke-width:4" />
  <circle cx="300" cy="400" r="50" style="fill:red; stroke:navy; stroke-width:2; stroke-dasharray: 5 2" />
  <ellipse cx="450" cy="200" rx="50" ry="100" style="fill:#197265; stroke:none"/>
  <polyline points="100 100 150 100 150 150 200 150 200 200 250 200 250 250 300 250 300 300 350 300 350 350
                    400 350 400 400 450 400 450 450 500 450 500 500"
            style="stroke:lime; stroke-width: 10; stroke-linejoin: round; stroke-linecap:round; stroke-opacity:50%" />
  <polygon points="50 100 100 50 500 50 550 100 550 500 500 550 100 550 50 500"
            style="stroke:aqua; stroke-width: 15; stroke-linejoin:round" />
  <line x1="100" x2="500" y1="500" y2="100"
            style="stroke:rgb(255,255,0); stroke-width:8; stroke-dasharray: 5 5 10 5 15 5 10 5 " />
</svg>
```





```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg SYSTEM "svg-19990812.dtd">
<svg width="600" height="600">
  <g style="fill:#aa7788; stroke:black; stroke-width:2; stroke-linejoin:round">
    <path d="M 150 250 L 150 150 A 100 100 0 0 0 250 150 A 100 100 0 0 1 350 150
             L 350 250 A 100 100 0 0 0 250 250 A 100 100 0 0 1 150 250"/>
  </g>
  <g style="fill:#77aa88; stroke:black; stroke-width:2; stroke-linejoin:round">
    <path d="M 150 450 L 150 350 A 50 50 0 0 0 250 350 A 50 50 0 0 1 350 350
             L 350 450 A 50 50 0 0 0 250 450 A 50 50 0 0 1 150 450"/>
  </g>
</svg>
```




```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG December 1999//EN"
  "http://www.w3.org/Graphics/SVG/SVG-19991203.dtd">
<svg width="400px" height="120px">
  <desc>Example Skew - Show effects of skewX and skewY</desc>
  <g style="fill:none; stroke:black; stroke-width:3">
    <!-- Draw the axes of the original coordinate system -->
    <line x1="0" y1="1.5" x2="400" y2="1.5" />
    <line x1="1.5" y1="0" x2="1.5" y2="120" />
  </g>
  <!-- Establish a new coordinate system whose origin is at (30,30)
       in the initial coord. system and which is skewed in X by 30 degrees. -->
  <g transform="translate(30,30)">
    <g transform="skewX(30)">
      <g style="fill:none; stroke:red; stroke-width:3">
        <line x1="0" y1="0" x2="50" y2="0" />
        <line x1="0" y1="0" x2="0" y2="50" />
      </g>
      <text x="0" y="0" style="font-size:20; font-family:Verdana; fill:blue">
        ABC (skewX)
      </text>
    </g>
  </g>
  <!-- Establish a new coordinate system whose origin is at (200,30)
       in the initial coord. system and which is skewed in Y by 30 degrees. -->
  <g transform="translate(200,30)">
    <g transform="skewY(30)">
      <g style="fill:none; stroke:red; stroke-width:3">
        <line x1="0" y1="0"
              x2="50" y2="0" />
        <line x1="0" y1="0"
              x2="0" y2="50" />
      </g>
      <text x="0" y="0"
            style="font-size:20;
            font-family:Verdana;
            fill:blue">
        ABC (skewY)
      </text>
    </g>
  </g>
</svg>
```

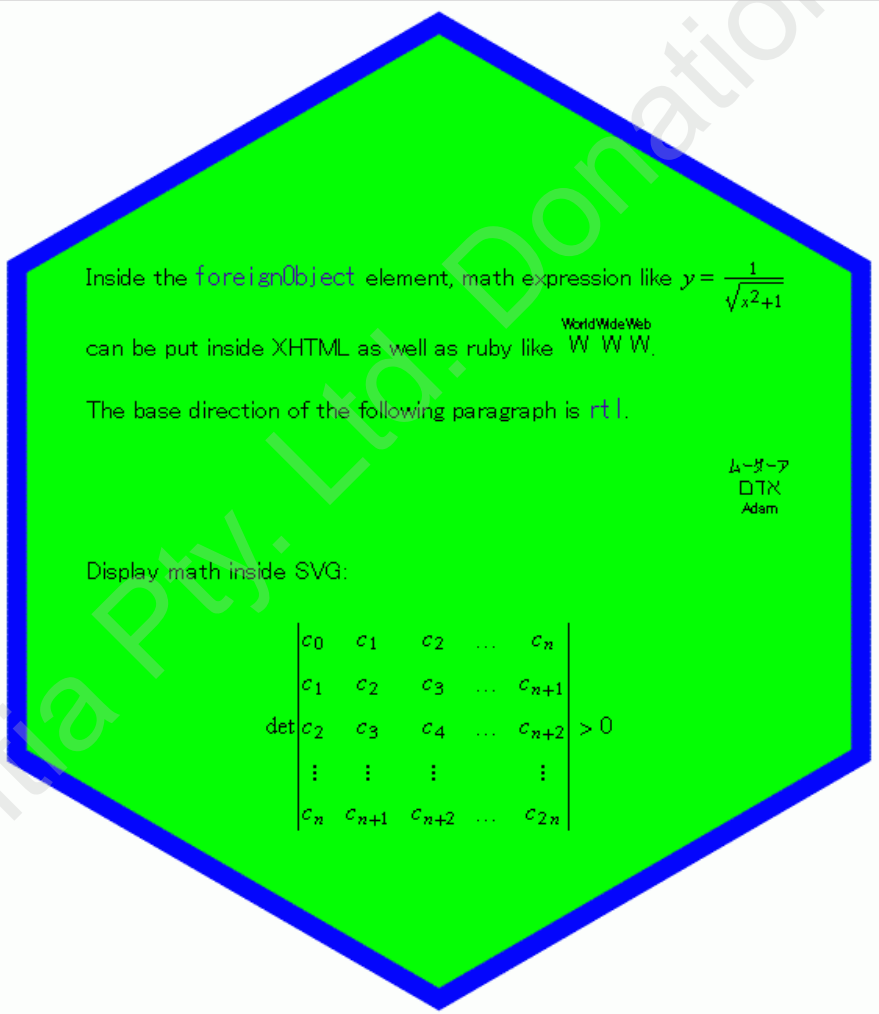
```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG December 1999//EN"
  "http://www.w3.org/Graphics/SVG/SVG-19991203.dtd">
<svg width="4in" heigt="3in">
  <defs style="stroke:green">
    <!-- Note that parent's stroke:green will have no effect below -->
    <circle id="TemplateObject02" cx="50" cy="50" r="30" style="fill:red" />
  </defs>
  <desc>
    Examples of <use> property inheritance
  </desc>
  <g style="fill:yellow;stroke:blue" >
    <!-- Draws a circle with fill:red and stroke:blue. -->
    <!-- Note that the referenced element specifies fill:red,
         which takes precedence over the inherited fill:yellow. -->
    <use xlink:href="#TemplateObject02" />
  </g>
</svg>
```

And More...

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG December 1999//EN"
  "http://www.w3.org/Graphics/SVG/SVG-19991203.dtd">
<svg width="4in" height="3in">
  <desc>
    This example shows the scripting capabilities of svg
  <desc>
  <defs>
    <script><![CDATA[
      /* Beep on mouseclick */
      MouseClickHandler() { beep(); }
     ]]>
    </script>
  </defs>
  <circle onclick="MouseClickHandler()" r="85"/>
</svg>
```

```
<?xml version="1.0" standalone="yes"?>
<svg width="4in" height="3in"
 xmlns = 'http://www.w3.org/Graphics/SVG/SVG-19991203.dtd'>
  <desc>
    This well formed svg document draws a triangle which is a hyperlink
  </desc>
  <a xmlns:xlink="http://www.w3.org/XML/XLink/0.9"
     xlink:type="simple" xlink:show="replace" xlink:actuate="user"
     xlink:href="http://www.w3.org">
    <path d="M 0 0 L 200 0 L 100 200 z"/>
  </a>
</svg>
```

W3C example of the XHTML+MathML+SVG profile



And More...

Tools

- **Much work is being done on producing SVG tools**
  - Adobe
    - *(15 Feb, 2000) SVG display plug-in for Netscape 4.x (Mac and Win), and SVG display ActiveX control for MS IE5 and MS Word (Win)*
    - *Illustrator, GoLive support*
  - Apache Batik SVG toolkit
  - Corel
    - *beta SVG export filter for CorelDraw*
  - CSIRO tools
  - Mayura Draw application
  - IBM server-side transcoders (CGI, AFP)
  - Flash → SVG converters

More at: *http://www.w3.org/Graphics/SVG/*

Vector Markup Language

- **A Microsoft scheme introduced with IE4**
  - support from Autodesk, Hewlett-Packard, Macromedia, and VISIO
  - also supported in Microsoft Office 2000

```
<html>
  <head>
    <title>Structured Graphic</title>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
      function Rotate(degVar){
        star.rotate(0,degVar,0);
        window.setTimeout("Rotate(5)",040,"JavaScript");
      }
      -->
    </script>
  </head>
  <body>
    <H1>Animated Yin-Yang</H1>
    <p>
    <object ID="star"
            CLASSID="CLSID:369303C2-D7AC-11D0-89D5-00A0C90833E6"
            STYLE="height: 500; width:500; zindex:10">
    <PARAM NAME="Line0001" VALUE="SetLineColor(0,0,0)">
    <PARAM NAME="Line0002" VALUE="SetFillColor(255,0,0)">
    <PARAM NAME="Line0003" VALUE="SetLineStyle(6)">
    <PARAM NAME="Line0004" VALUE="Polygon(102,8,17,-1,9,…,17,27)">
    <PARAM NAME="Line0005" VALUE="SetFillColor(255,255,0)">
    <PARAM NAME="Line0006" VALUE="Polygon(102,-9,-17,0, …,-18,-26)">
    </object>
    <p>
    <INPUT TYPE=button ID=rotate VALUE="ROTATE" onclick="Rotate(0)">
  </body>
</html>
```

Another VML Example...

```
<html xmlns:vml="urn:schemas-microsoft-com:vml">

  <head>
    <title>Example 22-2 from the XML Bible</title>
    <object id="VMLRender"
      classid="CLSID:10072CEC-8CC1-11D1-986E-00A0C955B42E">
    </object>
    <style>
      vml\:* { behavior: url(#VMLRender) }
    </style>
  </head>

  <body>
    <h1>Example 22-2 from the XML Bible</h1>

    <vml:shapetype id="fred"
       coordsize="21600,21600"
       fillcolor="blue"
       path="m@0,0l0,21600,21600,21600xe">
      <vml:formulas>
        <vml:f eqn="val #0"/>
        <vml:f eqn="prod #0 1 2"/>
        <vml:f eqn="sum @1 10800 0"/>
      </vml:formulas>
    </vml:shapetype>

    <vml:shape type="#fred" style="width:50px; height:50px" />
    <vml:shape type="#fred" style="width:100px; height:100px"/>
    <vml:shape type="#fred" style="width:150px; height:150px"/>

    <hr></hr>
    Last Modified March 23, 1999<br />
    Copyright 1999
    <a href="http://www.macfaq.com/personal.html">
      Elliotte Rusty Harold
    </a>
  </body>

</html>
```