**Unit III**

**HTML5 – SVG**

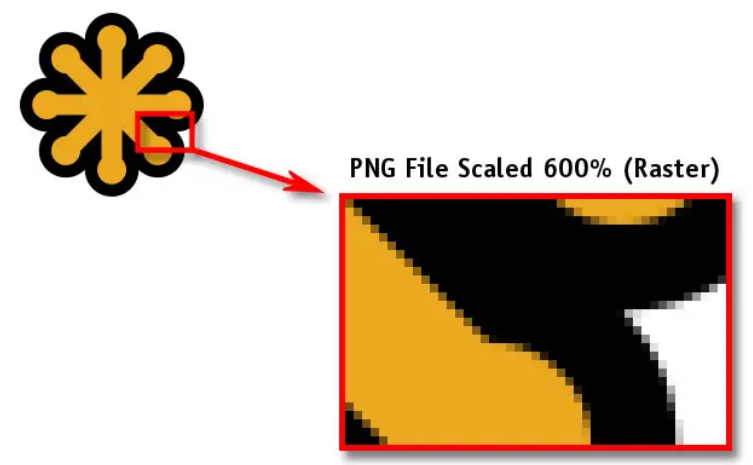
**HTML5 – SVG**: Viewing SVG Files, Embedding SVG in HTML5, HTML5 − SVG Circle, HTML5 − SVG Rectangle, HTML5 − SVG Line, HTML5 − SVG Ellipse, HTML5 − SVG Polygon, HTML5 − SVG Polyline, HTML5 − SVG Gradients, HTML5 − SVG Star.

**3.1 Introduction**

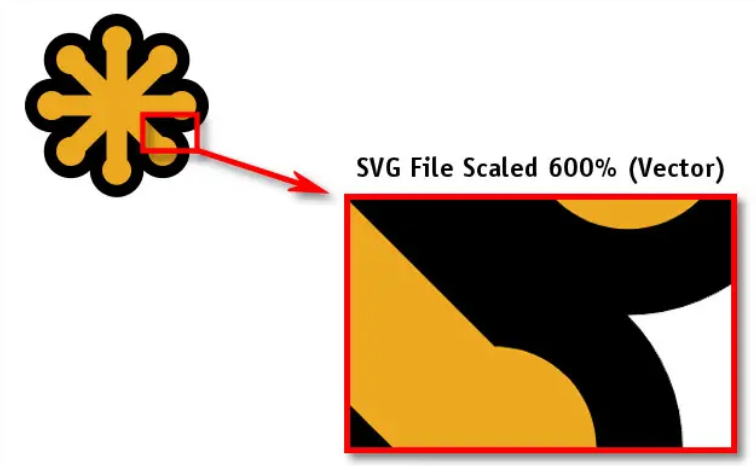
SVG stands for Scalable Vector Graphics. SVG is used to define vector-based graphics for the Web. It defines the graphics in XML format. We can animate every element and every attribute in SVG files.

An SVG file is a computer file that uses the SVG standard, [defined by the World Wide Web Consortium](https://www.w3.org/Graphics/SVG/), to display an image.

Typical image formats—such as JPG, PNG, and GIF—utilize raster graphics, which means that they display an image by storing a grid of pixels, called a bitmap. Each pixel’s color and location are stored in the file. If you scale a raster image larger than intended, the image becomes jagged and blurry.



In contrast, SVG images use vector graphics to define an image. In a vector graphics image, the [image is stored mathematically as a series of instructions](https://www.howtogeek.com/32597/whats-the-difference-between-pixels-and-vectors/) that tell a viewing program how to “draw” the image on your screen. Since the drawing can take place at any size, SVG files are resolution independent. They can be scaled bigger or smaller without losing quality or sharpness.



Since SVG files can be scaled to any size without detail loss, they can produce very smooth graphics with a relatively small file size compared to raster graphics. They are also more future-proof than bitmap images since they can be scaled to larger resolutions in future displays without losing quality.

Also, SVG is an [open standard](https://www.w3.org/Graphics/SVG/WG/wiki/SVG_FAQ) that can be included in viewing applications and browsers royalty-free, which means that application developers don’t need to pay any money to use it. SVG utilizes [XML](https://www.howtogeek.com/357092/what-is-an-xml-file-and-how-do-i-open-one/), another open standard, to encode two-dimensional graphics in a way that’s easy for apps to read and modify.

**Features of SVG**

* SVG is a W3C recommendation
* SVG integrates with other W3C standards such as the DOM and XSL
* SVG 1.0 became a W3C Recommendation on 4 September 2001.
* SVG 1.1 became a W3C Recommendation on 14 January 2003.
* SVG 1.1 (Second Edition) became a W3C Recommendation on 16 August 2011.

**SVG Advantages**

Advantages of using SVG over other image formats (like JPEG and GIF) are

* SVG images can be created and edited with any text editor
* SVG images can be searched, indexed, scripted, and compressed
* SVG images are scalable
* SVG images can be printed with high quality at any resolution
* SVG graphics do NOT lose any quality if they are zoomed or resized
* SVG is an open standard
* SVG files are pure XML

**SVG Shapes**

SVG has some predefined shape elements that can be used by developers, such as

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

**Supported Browsers**

* Google Chrome 4.0
* Internet Explorer 9.0
* Firefox 3.0
* Opera 10.1
* Safari 3.2

**3.2 Viewing SVG Files**

All modern web browsers includes Chrome, Edge, Firefox, and Safari support viewing SVG files. We can open a SVG file by **open** option of **file** menu.

Alternately ,we can use a free online converter such as [**SVGtoPNG.com**](https://www.svgtopng.com/) to convert an SVG file to a raster format that we can view using conventional image viewing applications. However, once we convert the SVG to a raster format like PNG, we will lose the ability to scale the image to different sizes while retaining the same sharpness and detail.

## Creation of an SVG File

To create or edit an SVG file, we need an application that supports the format. **Adobe Illustrator**, known for creating vector-based artwork, natively [supports both loading and saving files in SVG format](https://adobe.prf.hn/click/camref:1011lreog/pubref:htg710334%7Cxid%3Afr1688788714aaa/destination:https%3A%2F%2Fhelpx.adobe.com%2Fillustrator%2Fusing%2Fsvg.html).

Also,[**Inkscape**](https://inkscape.org/) and [**GIMP**](https://www.gimp.org/) are two free programs that can save artwork in SVG format. For other programs such as **Adobe Photoshop** and **InDesign**, we can purchase a plug-in that supports SVG, such as [**SVG Kit for Adobe Creative Suite**](https://redirect.viglink.com/?key=e7eab128eb8d1c53e14db14f4c632447&u=https%3A%2F%2Fsvg.scand.com%2F&cuid=xid:fr1688788714aab)**.**

**3.3 Embedding SVG in HTML5**

We can embed SVG elements directly into our HTML pages. SVG graphics are supported by the **<svg>** element in HTML. SVG graphics feature a container in which we can create a variety of shapes such as boxes, pathways, text, graphic images, and circles.

It takes the general format,

***<svg width="x" height="y">***

***shape\_tags...***

***</svg>***

where,

**width & height**in the rectangle represents the height and width of the rectangle

For example,

<svg width="100" height="100">

<circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />

</svg>

**Some other attributes**

* **viewBox** attribute denotes the dimension of an SVG viewport. It attribute indicates the center of the coordinate on which the elements of the image are positioned.
* **stroke attribute** represents the color to be used to paint the outline of the shape.
* **fill attribute**denotes the color to be used to paint the element.
* **stroke-width** represents the width of the color to use for the shape.
* **rgb(x,y,z):** This is a triplet of red, green, and blue (RGB) values to deﬁne how much of each primary color is used to generate the desired color.
* **contentScriptType** attribute specifies a default scripting language for the given document fragment on the <svg> element.
* **contentStyleType**attribute represents the style sheet language for the given document fragment.
* **preserveAspectRatio**attribute denotes how an element with a viewBox providing a given aspect ratio must fit into a viewport with a different aspect ratio.
* **version**attribute is used to indicate the specification of a document type SVG.

**3.4** **HTML5 − SVG Circle**

The **<circle>** element is used to create a circle. **<circle>** element draws a circle on the screen which is done by the 3 basic parameters which include **cx, cy,**and **r**. It takes the general format,

***<circle***

***cx="x-axis co-ordinate"***

***cy="y-axis co-ordinate"***

***r="length" >***

***</circle>***

where

**cx 🡪**  x-axis co-ordinate of the center of the circle. The default value is 0.

**cy 🡪**y-axis co-ordinate of the center of the circle.  The default value is 0.

**r 🡪**  Radius of the circle.

**cx** and**cy** is the x-axis and y-axis co-ordinate which determines theposition of the circle and r is the radius of the circle which determines the **size** of the circle. Circle rendering will be disabled if its radius is zero.

**Example 3.1 SVG Circle**

<!DOCTYPE html>

<html>

<body>

<h1>My first SVG</h1>

<svg width="100" height="100">

<circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />

</svg>

</body>

</html>

**3.5 HTML5 − SVG Rectangle**

The <rect> element is used to create a rectangle and variations of a rectangle shape. It takes the general format,

***<rect***

***x="****x-axis co-ordinate****"***

***y="****y-axis co-ordinate****"***

***width="****length****"***

***height="l****ength****"***

***rx="****length****"***

***ry="****length****"***

***style="****style information****"***

***class="****style class****" >***

***</rect>***

where

**X🡪** x-axis co-ordinate of top left.

**Y🡪** y-axis co-ordinate of top left.

**width🡪**Width of rectangle.

**height🡪** Height of rectangle.

**rx🡪** Roundness of the x-axis.

**ry🡪**Roundness of the y-axis.

**style 🡪** Specify the inline style.

**class 🡪** Specify the external style.

**Example 3.2 SVG Rectangle**

<html>

<body>

<h1>SVG Rectangle</h1>

<svg width="400" height="110">  
  <rect width="300" height="100" style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />  
</svg>

</body>

</html>

**Example 3.3 Square in different color and stroke**

<html>

<body>

<h1>SVG Square </h1>

<svg width="400" height="180">

<rect x="50" y="20" width="150" height="150"

style="fill:blue;stroke:pink;stroke-width:5;fill-opacity:0.1;stroke-opacity:0.9" />

</svg>

</body>

</html>

**Example 3.4 Square in different stroke**

<html>

<body>

<h1>SVG Square </h1>

<svg width="400" height="180">

<rect x="50" y="20" width="150" height="150"

style="fill:blue;stroke:pink;stroke-width:5;opacity:0.5" />

</svg>

</body>

</html>

**Example 3.5 Rounded Rectangle**

<html>

<body>

<h1>SVG Rectangle</h1>

<svg width="400" height="180">

<rect x="50" y="20" rx="20" ry="20" width="150" height="150"

style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />

</svg>

</body>

</html>

**3.6 HTML5 − SVG Line**

The <line> element is used to create a line.

***<line***

***x1="****x-axis co-ordinate****"***

***y1="****y-axis co-ordinate****"***

***x2="****x-axis co-ordinate****"***

***y2="****y-axis co-ordinate****"***

***style="****style information****"***

***class="****style class****" >***

***</line>***

where

**X1🡪** x-axis co-ordinate of starting point

**Y1🡪** y-axis co-ordinate of starting point

**X2🡪** x-axis co-ordinate of finishing point

**Y2🡪** y-axis co-ordinate of finishing point

**style 🡪** Specify the inline style.

**class 🡪** Specify the external style.

**Example 3.6 SVG Line**

<html>

<body>

<h1>SVG Rectangle</h1>

<svg height="210" width="500">

<line x1="0" y1="0" x2="200" y2="200" style="stroke:rgb(255,0,0);stroke-width:2" />

</svg>

</svg>

</body>

</html>

**3.7 HTML5 − SVG Ellipse**

The <ellipse> element is used to create an ellipse. An ellipse is closely related to a circle. The difference is that an ellipse has an x and a y radius that differs from each other, while a circle has equal x and y radius. It takes the general format,

***<ellipse***

***cx="x-axis co-ordinate"***

***cy="y-axis co-ordinate"***

***rx="length"***

***ry="length" >***

***</ellipse>***

where,

**cx🡪** x-axis co-ordinate of the center

**cy🡪** y-axis co-ordinate of the center

**rx🡪**x-axis radius of the ellipse

**ry🡪** y-axis radius of the ellipse

**Example 3.7 SVG Ellipse**

<html>

<body>

<h1>SVG **Ellipse** </h1>

<svg height="140" width="500">

<ellipse cx="200" cy="80" rx="100" ry="50"

style="fill:yellow;stroke:purple;stroke-width:2" />

</svg>

</body>

</html>

**Example 3.8 SVG Ellipse**

<html>

<body>

<h1>SVG Ellipse</h1>

<svg height="150" width="500">

<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>

</body>

</html>

**Example 3.9 SVG Ellipse**

<html>

<body>

<h1>SVG Ellipse</h1>

<svg height="150" width="500">

<ellipse cx="240" cy="50" rx="220" ry="30" style="fill:yellow" />

<ellipse cx="220" cy="50" rx="190" ry="20" style="fill:white" />

</svg>

</body>

</html>

**3.8 HTML5 − SVG Polygon**

The <polygon> element is used to create a graphic that contains at least three sides. Polygons are made of straight lines, and the shape is "closed" (all the lines connect up). Polygon word is derived from Greek. **Poly** means many and **gon** means angle.

The **<polygon>**element of SVG is used to make any type of polygon on the SVG and defines a closed shape consisting of a set of connected straight line segments. It takes the general format,

***<polygon points="Pair of points required to draw the shape"***

***stroke="stroke color"***

***fill="fill color for colored closed shapes">***

where,

**points🡪**Pair of points required to draw the shape

**pathLength🡪**It tells the total length of the path.

**Example 3.10 SVG Polygon**

<html>

<body>

<h1>SVG polygon</h1>

<svg height="210" width="500">

<polygon points="200,10 250,190 160,210" style="fill:lime;stroke:purple;stroke-width:1" />

</svg>

</body>

</html>

**Example 3.11 SVG Polygon**

<html>

<body>

<h1>SVG polygon</h1>

<svg height="250" width="500">

<polygon points="220,10 300,210 170,250 123,234" style="fill:lime;stroke:purple;stroke-width:1" />

</svg>

</body>

</html>

**3.9 HTML5 − SVG Polyline**

The**<polyline>**element of SVG in HTML is used to create a shape by connecting lines through different points. It can be used to create open shapes. It takes the general format,

***<polyline points="***Pair of points required to draw the shape***"***

***stroke="***stroke color***"***

***fill="***fill color for colored closed shapes***">***

where

**points 🡪**Pair of points required to draw the shape

**pathLength 🡪**It tells the total length of the path

**Example 3.12 SVG Polyline**

<html>

<body>

<h1>SVG polyline</h1>

<svg height="200" width="500">

<polyline points="20,20 40,25 60,40 80,120 120,140 200,180"

style="fill:none;stroke:black;stroke-width:3" />

</svg>

</body>

</html>

**Example 3.13 SVG Polyline without filling**

<html>

<body>

<h1>SVG polyline</h1>

<svg height="180" width="500">

<polyline points="0,40 40,40 40,80 80,80 80,120 120,120 120,160"

style="fill:none;stroke:red;stroke-width:4" />

</svg>

</body>

</html>

**Example 3.14 SVG Polyline with filling**

<html>

<body>

<h1>SVG polyline</h1>

<svg height="180" width="500">

<polyline points="0,40 40,40 40,80 80,80 80,120 120,120 120,160"

style="fill:blue;stroke:red;stroke-width:4" />

</svg>

</body>

</html>

**3.10 HTML5 − SVG Gradients**

A gradient is a smooth transition from one color to another. In addition, several color transitions can be applied to the same element. There are two main types of gradients in SVG:

* Linear
* Radial

# SVG Gradients - Linear

The <linearGradient> element is used to define a linear gradient to other elements. It allows a smooth transition from one color to another. This is the most reliable technique.

The <linearGradient> element must be nested within a <defs> tag. The <defs> tag is short for definitions and contains definition of special elements (such as gradients).

Linear gradients can be defined as horizontal, vertical or angular gradients:

* Horizontal gradients are created when y1 and y2 are equal and x1 and x2 differ
* Vertical gradients are created when x1 and x2 are equal and y1 and y2 differ
* Angular gradients are created when x1 and x2 differ and y1 and y2 differ

[**Attributes**](https://developer.mozilla.org/en-US/docs/Web/SVG/Element/linearGradient#attributes)

[**gradientUnits**](https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/gradientUnits)

This attribute defines the coordinate system for attributes x1, x2, y1, y2 Value type: userSpaceOnUse|objectBoundingBox ;

Default value: objectBoundingBox;

Animatable: **yes**

[**gradientTransform**](https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/gradientTransform)

This attribute provides additional [transformation](https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/transform) to the gradient coordinate system.

Value type: [<transform-list>](https://developer.mozilla.org/en-US/docs/Web/SVG/Content_type#transform-list) ;

Default value: identity transform;

 Animatable: **yes**

[**href**](https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/href)

This attribute defines a reference to another <linearGradient> element that will be used as a template.

Value type:[**<URL>**](https://developer.mozilla.org/en-US/docs/Web/SVG/Content_type#url) ;

 Default value: none;

Animatable: **yes**

[**spreadMethod**](https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/spreadMethod)

This attribute indicates how the gradient behaves if it starts or ends inside the bounds of the shape containing the gradient.

Value type: pad|reflect|repeat ;

Default value: pad;

 Animatable: **yes**

[**x1**](https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/x1)

This attribute defines the x coordinate of the starting point of the vector gradient along which the linear gradient is drawn.

 Value type: [<length-percentage>](https://developer.mozilla.org/en-US/docs/Web/CSS/length-percentage) | [<number>](https://developer.mozilla.org/en-US/docs/Web/CSS/number);

 Default value: 0%;

Animatable: **yes**

[**x2**](https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/x2)

This attribute defines the x coordinate of the ending point of the vector gradient along which the linear gradient is drawn.

Value type: [<length-percentage>](https://developer.mozilla.org/en-US/docs/Web/CSS/length-percentage) | [<number>](https://developer.mozilla.org/en-US/docs/Web/CSS/number);

Default value: 100%;

Animatable: **yes**

[**y1**](https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/y1)

This attribute defines the y coordinate of the starting point of the vector gradient along which the linear gradient is drawn.

 Value type: [<length-percentage>](https://developer.mozilla.org/en-US/docs/Web/CSS/length-percentage) | [<number>](https://developer.mozilla.org/en-US/docs/Web/CSS/number);

 Default value: 0%;

Animatable: **yes**

[**y2**](https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/y2)

This attribute defines the y coordinate of the ending point of the vector gradient along which the linear gradient is drawn.

 Value type: [<length-percentage>](https://developer.mozilla.org/en-US/docs/Web/CSS/length-percentage) | [<number>](https://developer.mozilla.org/en-US/docs/Web/CSS/number);

Default value: 0%;

Animatable: **yes**

**Example 3**.**15**  **Define an ellipse with a horizontal linear gradient from yellow to red**

<html>

<body>

<h1> SVG Linear Gradient </h1>

<svg height="150" width="400">

<defs>

<linearGradient id="grad1" 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="70" rx="85" ry="55" fill="url(#grad1)" />

</svg>

</body>

</html>

**Example 3.16 Define an ellipse with a vertical linear gradient from yellow to red**

<html>

<body>

<h1> SVG Linear Gradient </h1>

<svg height="150" width="400">

<defs>

<linearGradient id="grad1" 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="70" rx="85" ry="55" fill="url(#grad1)" />

</svg>

</body>

</html>

**Example 3.17 Define an ellipse with a horizontal linear gradient from yellow to red**

<html>

<body>

<h1> SVG Linear Gradient </h1>

<svg height="150" width="400">

<defs>

<linearGradient id="grad2" x1="0%" y1="0%" x2="0%" y2="100%">

<stop offset="0%" style="stop-color:rgb(255,0,0);stop-opacity:1" />

<stop offset="100%" style="stop-color:rgb(255,255,0);stop-opacity:1" />

</linearGradient>

</defs>

<ellipse cx="200" cy="70" rx="85" ry="55" fill="url(#grad2)" />

</svg>

</body>

</html>

### Example 3.18 Define an ellipse with a horizontal linear gradient from yellow to red, and add a text inside the ellipse

<html>

<body>

<h1> SVG Linear Gradient </h1>

<svg height="1450" width="2000">

<defs>

<linearGradient id="grad3" 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="350" cy="200" rx="250" ry="120" fill="url(#grad3)" />

<text fill="#ffffff" font-size="45" font-family="Verdana " x="200" y="186">

Royal BCA</text>

</svg>

</body>

</html>

## SVG Radial Gradient

The <radialGradient> element is used to define a radial gradient. The <radialGradient> element must be nested within a <defs> tag. The <defs> tag is short for definitions and contains definition of special elements (such as gradients).

**Example3.19 Define an ellipse with a radial gradient from white to blue**

<html>

<body>

<h1> SVG Radial Gradient </h1>

<svg height="150" width="500">

<defs>

<radialGradient id="grad1" cx="50%" cy="50%" r="50%" fx="50%" fy="50%">

<stop offset="0%" style="stop-color:rgb(255,255,255);

stop-opacity:0" />

<stop offset="100%" style="stop-color:rgb(0,0,255);stop-opacity:1" />

</radialGradient>

</defs>

<ellipse cx="200" cy="70" rx="85" ry="55" fill="url(#grad1)" />

</svg>

</body>

</html>

**3.11 HTML5 − SVG Star**

## We can make star by using polygon as follows.

**Example 3.20 SVG Polygon - STAR**

<html>

<body>

<h1>SVG polygon</h1>

<svg height="210" width="500">

<polygon points="100,10 40,198 190,78 10,78 160,198"

style="fill:lime;stroke:purple;stroke-width:5;fill-rule:nonzero;" />

</svg>

</body>

</html>

**Example 3.21 SVG Polygon - STAR**

<html>

<body>

<h1>SVG polygon</h1>

<svg height="210" width="500">

<polygon points="100,10 40,198 190,78 10,78 160,198"

style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />

</svg>

</body>

</html>

## SVG Path - <path>

The <path> element is used to define a path.

The following commands are available for path data:

* M = moveto
* L = lineto
* H = horizontal lineto
* V = vertical lineto
* C = curveto
* S = smooth curveto
* Q = quadratic Bézier curve
* T = smooth quadratic Bézier curveto
* A = elliptical Arc
* Z = closepath

**Note:** All of the commands above can also be expressed with lower letters. Capital letters means absolutely positioned, lower cases means relatively positioned.

**Example 3.22 SVG Path**

The example below defines a path that starts at position 150,0 with a line to position 75,200 then from there, a line to 225,200 and finally closing the path back to 150,0

<html>

<body>

<h1>SVG path</h1>

<svg height="210" width="400">

<path d="M150 0 L75 200 L225 200 Z" />

</svg>

</body>

</html>

## SVG Text - <text>

The <text> element is used to define a text.

**Example 3.23 SVG Text**

<html>

<body>

<h1> SVG Text </h1>

<svg height="30" width="200">

<text x="0" y="15" fill="red">I love Myself!</text>

</svg>

</body>

</html>

**Example 3.24 SVG Text**

<html>

<body>

<h1> SVG Text </h1>

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

<text x="0" y="15" fill="red" transform="rotate(30 20,40)">Christy@KarthikaJesus</text>t>

</svg>

</body>

</html>