Unit3:

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.

SVG stands for **S**calable **V**ector **G**raphics and it is a language for describing 2D-graphics and graphical applications in XML and the XML is then rendered by an SVG viewer.

SVG is mostly useful for vector type diagrams like Pie charts, Two-dimensional graphs in an X,Y coordinate system etc.

SVG became a W3C Recommendation 14. January 2003 and you can check latest version of SVG specification at SVG Specification.

Viewing SVG Files

Most of the web browsers can display SVG just like they can display PNG, GIF, and JPG. Internet Explorer users may have to install the Adobe SVG Viewer to be able to view SVG in the browser.

Embedding SVG in HTML5

HTML5 allows embedding SVG directly using **<svg>...</svg>** tag which has following simple syntax −

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

Firefox 3.7 has also introduced a configuration option ("about:config") where you can enable HTML5 using the following steps −

- Type **about:config** in your Firefox address bar.
- Click the "I'll be careful, I promise!" button on the warning message that appears (and make sure you adhere to it!).
- Type **html5.enable** into the filter bar at the top of the page.
- Currently it would be disabled, so click it to toggle the value to true.

Now your Firefox HTML5 parser should be enabled and you should be able to experiment with the following examples.

HTML5 − SVG Circle

Following is the HTML5 version of an SVG example which would draw a circle using <circle> tag –

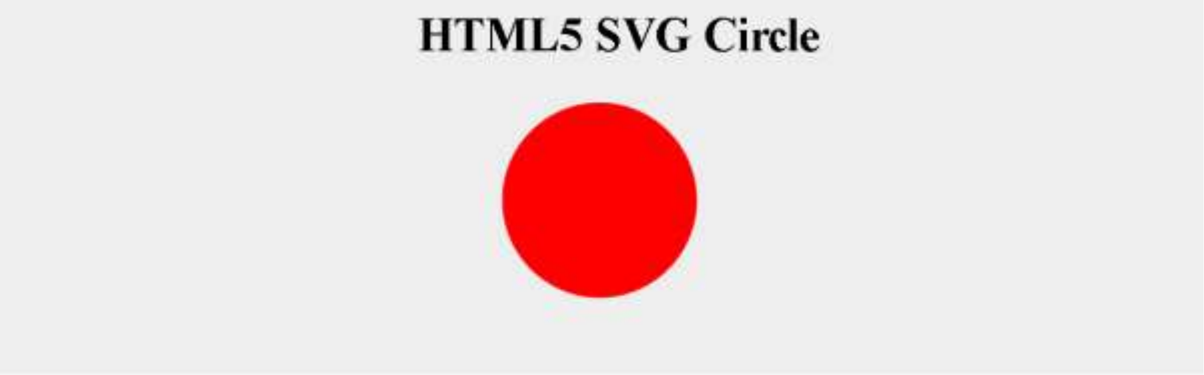
```
<!DOCTYPE html>

<html>
   <head>
```

```
    <title>SVG</title>
    <meta charset = "utf-8" />
  </head>

  <body>
    <h2 align = "center">HTML5 SVG Circle</h2>

    <svg id = "svgelem" height = "200" >
      <circle id = "redcircle" cx = "50" cy = "50" r = "50" fill = "red" />
    </svg>
  </body>
</html>
```

Code explanation:

- The cx and cy attributes define the x and y coordinates of the center of the circle. If cx and cy are omitted, the circle's center is set to (0,0)
- The r attribute defines the radius of the circle

This would produce the following result in HTML5 enabled latest version of Firefox.



HTML5 SVG Circle

HTML5 − SVG Rectangle

Following is the HTML5 version of an SVG example which would draw a rectangle using <rect> tag −

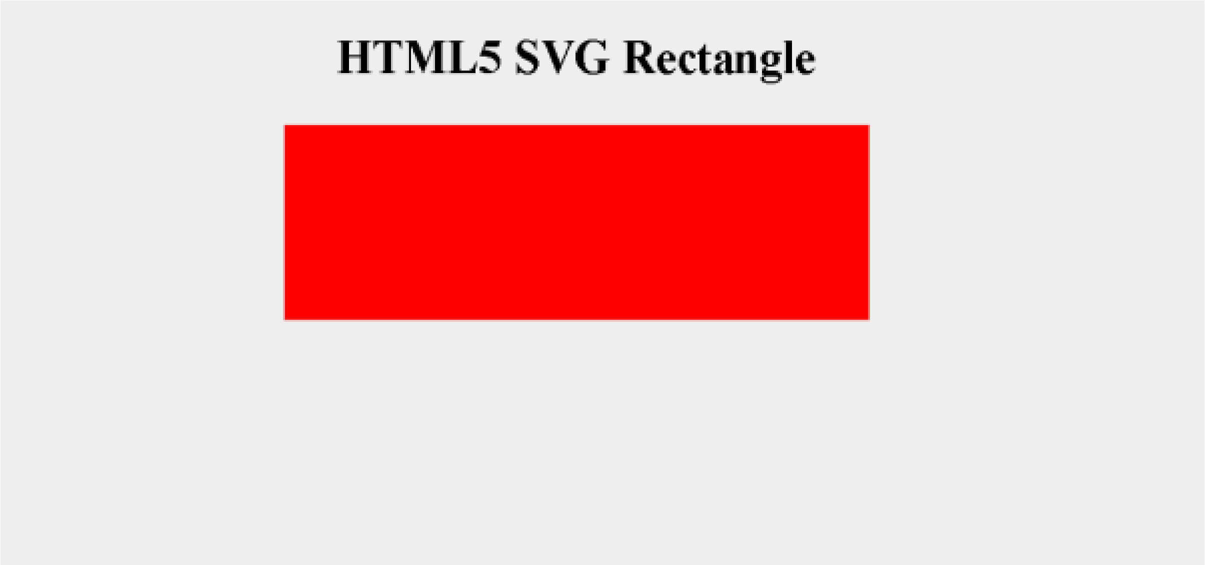
```
<!DOCTYPE html>

<html>
  <head>

    <title>SVG</title>
    <meta charset = "utf-8" />
  </head>

  <body>
    <h2 align = "center">HTML5 SVG Rectangle</h2>

    <svg id = "svgelem" height = "200">
```

```
      <rect id = "redrect" width = "300" height = "100" fill = "red"
style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />
    </svg>
  </body>
</html>
```

Code explanation:

- The width and height attributes of the <rect> element define the height and the width of the rectangle
- The style attribute is used to define CSS properties for the rectangle
- The CSS fill property defines the fill color of the rectangle
- The CSS stroke-width property defines the width of the border of the rectangle
- The CSS stroke property defines the color of the border of the rectangle

This would produce the following result in HTML5 enabled latest version of Firefox.


HTML5 SVG Rectangle

Example 2:

```
<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>
```

Code explanation:

- The x attribute defines the left position of the rectangle (e.g. x="50" places the rectangle 50 px from the left margin)
- The y attribute defines the top position of the rectangle (e.g. y="20" places the rectangle 20 px from the top margin)
- The CSS fill-opacity property defines the opacity of the fill color (legal range: 0 to 1)
- The CSS stroke-opacity property defines the opacity of the stroke color (legal range: 0 to 1)

Example 3:

```
<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>
```

Code explanation:

- The rx and the ry attributes rounds the corners of the rectangle

HTML5 − SVG Line

Following is the HTML5 version of an SVG example which would draw a line using <line> tag −

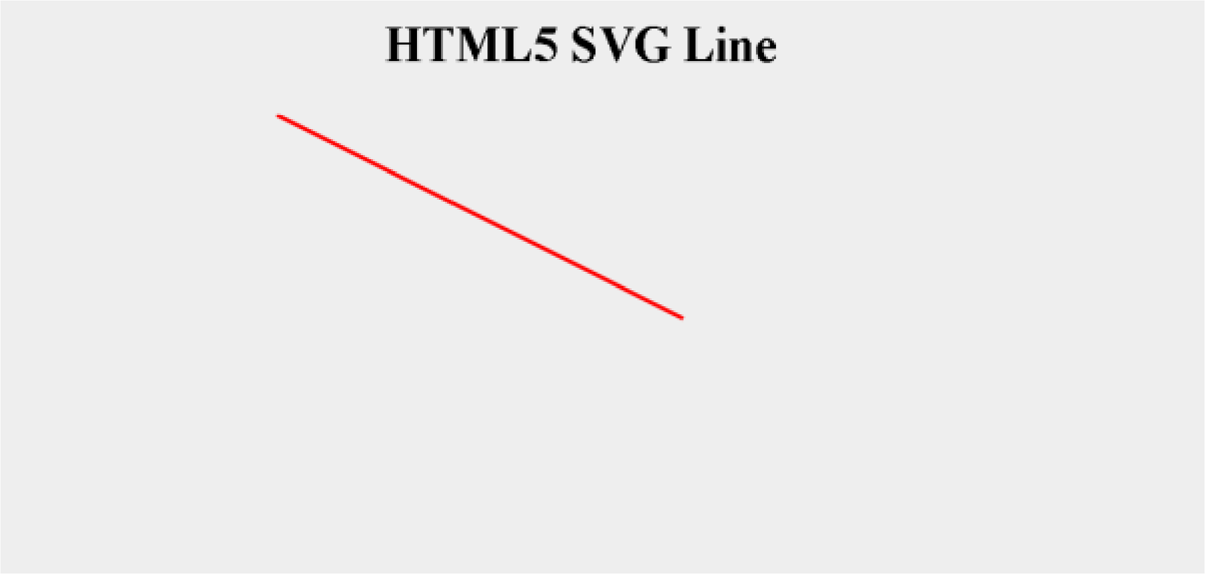
```
<!DOCTYPE html>

<html>
   <head>
         <title>SVG</title>
      <meta charset = "utf-8" />
   </head>

   <body>
      <h2 align = "center">HTML5 SVG Line</h2>

      <svg id = "svgelem" height = "200" xmlns = "http://www.w3.org/2000/svg">
         <line x1 = "0" y1 = "0" x2 = "200" y2 = "100"
            style = "stroke:red;stroke-width:2"/>
      </svg>
   </body>
</html>
```

You can use the **style** attribute which allows you to set additional style information like stroke and fill colors, width of the stroke, etc.

Code explanation:

- The x1 attribute defines the start of the line on the x-axis
- The y1 attribute defines the start of the line on the y-axis
- The x2 attribute defines the end of the line on the x-axis
- The y2 attribute defines the end of the line on the y-axis

This would produce the following result in HTML5 enabled latest version of Firefox.



HTML5 − SVG Ellipse

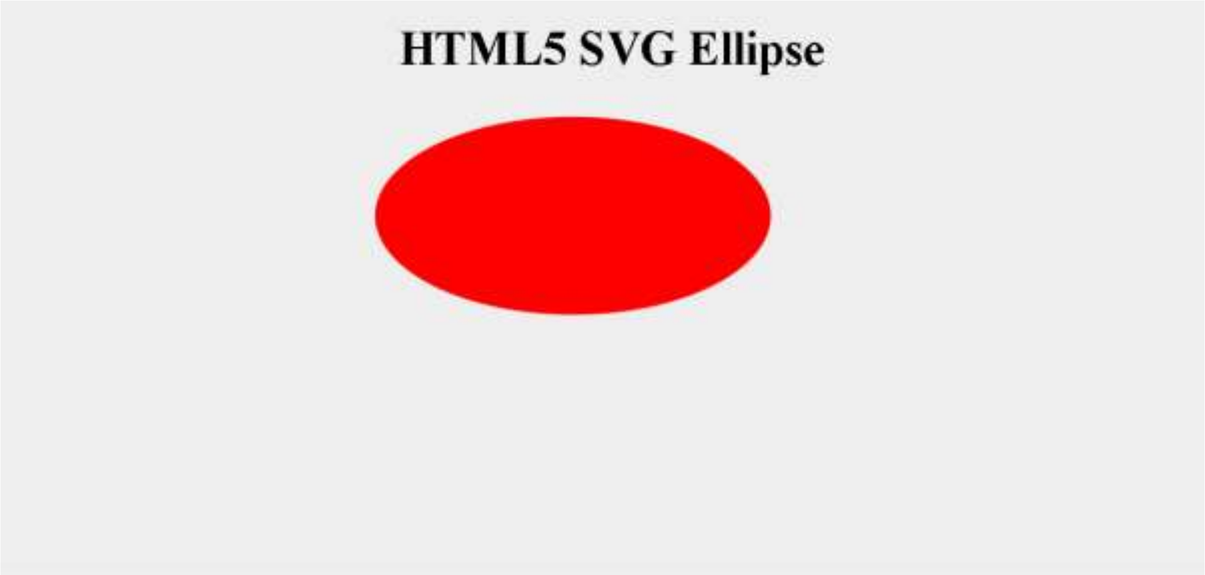
Following is the HTML5 version of an SVG example which would draw an ellipse using <ellipse> tag −

```
<!DOCTYPE html>

<html>
   <head>
      <title>SVG</title>
      <meta charset = "utf-8" />
   </head>

   <body>
      <h2 align = "center">HTML5 SVG Ellipse</h2>

      <svg id = "svgelem" height = "200">
         <ellipse cx = "100" cy = "50" rx = "100" ry = "50" fill = "red" />
      </svg>

   </body>
</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.

Code explanation:

- The cx attribute defines the x coordinate of the center of the ellipse
- The cy attribute defines the y coordinate of the center of the ellipse
- The rx attribute defines the horizontal radius
- The ry attribute defines the vertical radius



HTML5 SVG Ellipse

HTML5 − SVG Polygon

- A **polygon** is a two-dimensional geometric figure that has a finite number of sides.
- The sides of a polygon are made of straight line segments connected to each other end to end.
- Thus, the line segments of a polygon are called sides or edges.
- The point where two line segments meet is called vertex or corners,
- The simplest polygons are triangles (three sides), quadrilaterals (four sides), and pentagons (five sides).

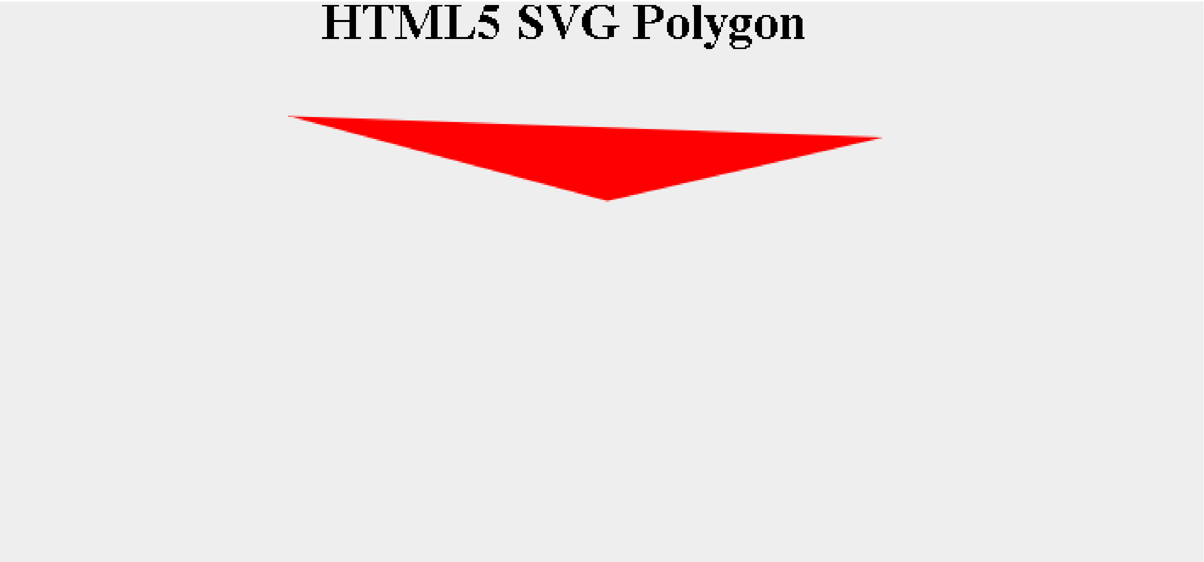
Following is the HTML5 version of an SVG example which would draw a polygon using <polygon> tag −

```
<!DOCTYPE html>

<html>
  <head>
    <title>SVG</title>
    <meta charset = "utf-8" />
  </head>

  <body>
    <h2 align = "center">HTML5 SVG Polygon</h2>

    <svg id = "svgelem" height = "200">
      <polygon  points = "20,10 300,20, 170,50" fill = "red" />
    </svg>
  </body>
</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.

Code explanation:

- The points attribute defines the x and y coordinates for each corner of the polygon



HTML5 SVG Polygon

HTML5 − SVG Polyline

A polyline is a connected sequence of line segments created as a single object.

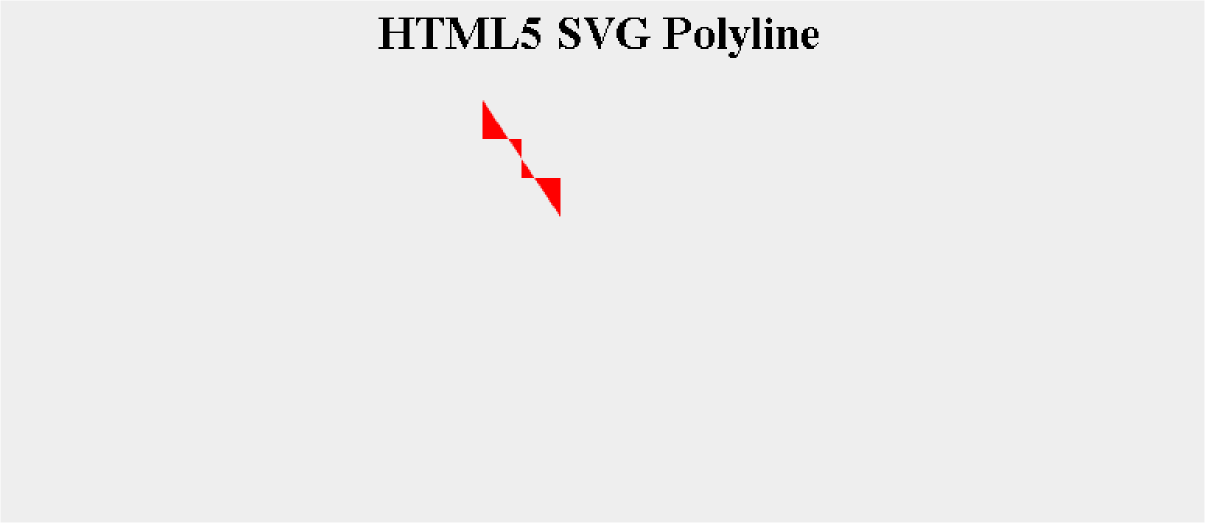
Following is the HTML5 version of an SVG example which would draw a polyline using <polyline> tag −

```
<!DOCTYPE html>

<html>
  <head>
    <title>SVG</title>
    <meta charset = "utf-8" />
  </head>

  <body>
    <h2 align = "center">HTML5 SVG Polyline</h2>
    <svg id = "svgelem" height = "200">
      <polyline points = "0,0 0,20 20,20 20,40 40,40 40,60" fill = "red" />
    </svg>
  </body>
</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.

Code explanation:

- The points attribute defines the list of points (pairs of x and y coordinates) required to draw the polyline



HTML5 SVG Polyline

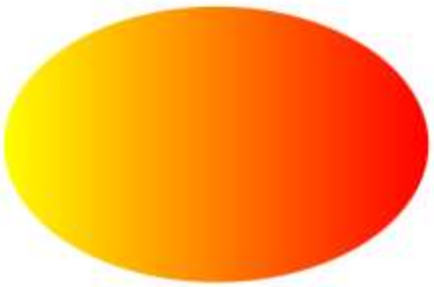
HTML5 − SVG Gradients

CSS gradients let you display smooth transitions between two or more specified colors

SVG Gradients

Example 1

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



Here is the SVG code:

```
<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>
```

Code explanation:

- The id attribute of the <linearGradient> tag defines a unique name for the gradient
- The x1, x2, y1,y2 attributes of the <linearGradient> tag define the start and end position of the gradient
- The color range for a gradient can be composed of two or more colors. Each color is specified with a <stop> tag. The offset attribute is used to define where the gradient color begin and end
- The fill attribute links the ellipse element to the gradient

SVG Linear Gradient - <linearGradient>

The <linearGradient> element is used to define a linear gradient.

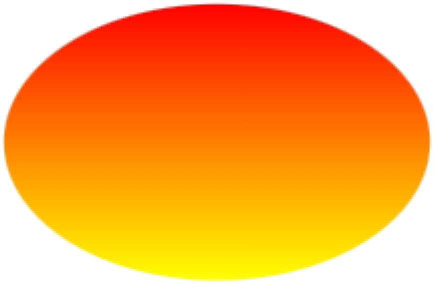
The <linearGradient> element must be nested within a <defs> tag. The <defs> tag is short for definitions and contains definition of special elements (such as gradients).

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

Example 2

Define an ellipse with a vertical linear gradient from yellow to red:



Here is the SVG code:

```
<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
```

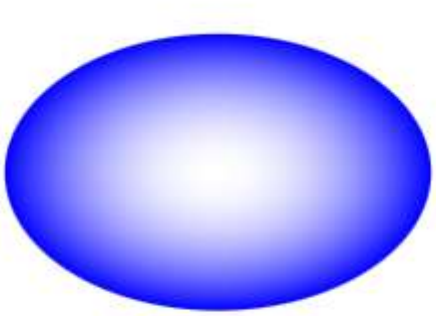
SVG radial Gradient - <radialGradient>

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).

Example 1

Define an ellipse with a radial gradient from white to blue:



```
<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>
```

Code explanation:

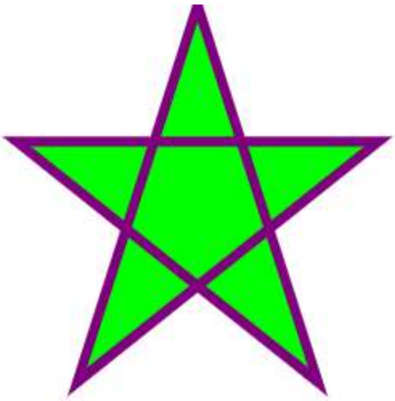
- The id attribute of the <radialGradient> tag defines a unique name for the gradient
- The cx, cy and r attributes define the outermost circle and the fx and fy define the innermost circle
- The color range for a gradient can be composed of two or more colors. Each color is specified with a <stop> tag. The offset attribute is used to define where the gradient color begin and end
- The fill attribute links the ellipse element to the gradient

HTML5 − SVG Star

Following is the HTML5 version of an SVG example which would draw a star using <polygon> tag.

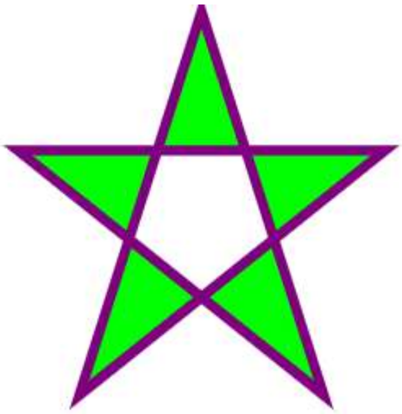
```
<html>
  <head>
    <title>SVG</title>
    <meta charset = "utf-8" />
  </head>

  <body>
    <h2 align = "center">HTML5 SVG Star</h2>

    <svg id = "svgelem" height = "200" >
      <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>
```

This would produce the following result in HTML5 enabled latest version of Firefox.



Change the fill-rule property to "evenodd":

```
<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>
```



- if i change **fill-rule="nonzero"** the entire SVG get the fill.
- Currently with **fill-rule="evenodd"** applied the SVG's central area does't get the fill.