

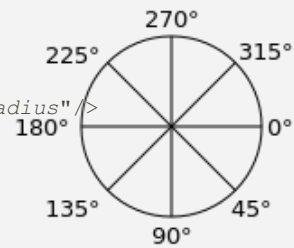
Basic Shapes

line	<code><line x1="start-x" y1="start-y" x2="end-x" y2="end-y"/></code>
rectangle	<code><rect x="left-x" y="top-y" width="width" height="height"/></code>
circle	<code><circle cx="center-x" cy="center-y" r="radius"/></code>
ellipse	<code><ellipse cx="center-x" cy="center-y" rx="x-radius" ry="y-radius"/></code>
polygon	<code><polygon points="points-list"/></code>
polyline	<code><polyline points="points-list"/></code>

Transformations

<code>translate(x, y)</code>	moves <i>x</i> horizontally, <i>y</i> vertically
<code>scale(xFactor, yFactor)</code>	multiplies by <i>xFactor</i> and <i>yFactor</i>
<code>scale(factor)</code>	equivalent to <code>scale(factor, factor)</code>
<code>rotate(angle, centerX, centerY)</code>	rotates by <i>angle</i> degrees with center of rotation (<i>centerX</i> , <i>centerY</i>)
<code>rotate(angle)</code>	equivalent to <code>rotate(angle, 0, 0)</code>
<code>skewX(angle)</code>	skews <i>x</i> -coordinates by <i>angle</i> degrees
<code>skewY(angle)</code>	skews <i>y</i> -coordinates by <i>angle</i> degrees
<code>matrix(a b c d e f)</code>	specifies a transformation matrix of six values

Angle Measurements



Angle measurements increase clockwise, starting from the positive *x*-axis.

Grouping and Referencing Objects

grouping	<code><g id="id" style="attributes"></code> <code></g></code>
use a group	<code><use xlink:href="#id" x="x1" y="y1"/></code>
defining groups without displaying	<code><defs></code> <code></defs></code>
symbol	<code><symbol id="id" style="attributes" preserveAspectRatio="xMidYMid meet" /></code>

Gradients

attributes

spreadMethod	pad
	repeat
	reflect
gradientTransform	skewX
	skewY
	rotate

`<linearGradient>` attributes

`x1 y1 x2 y2` = "0-100%"

`<radialGradient>` attributes

`cx cy r fx fy` = "0-100%"

elements

<code><stop></code>	offset="0-100%"
	stop-color:
	stop-opacity: 0.0-1.0

Clipping and masking

clipping

<clipPath> id, clipPathUnits

<use xlink:href="#imageid" style="clip-path: url(#pathid);"/>

masking

<mask> id, x, y, width, height

clipPathUnits objectBoundingBox, userSpaceOnUse

maskUnits

maskContentUnits

style mask: url(#maskid)

fill-opacity: 0.0-1.0

fill: color; white specified for opacity only

Filters

<filter> x, y, width, height

filterUnits objectBoundingBox, userSpaceOnUse

primitiveUnits

<feGaussianBlur> can create a drop shadow

in SourceAlpha, SourceGraphic

stdDeviation blur or x-blur y-blur

Stroke Attributes

style="attribute: value" specify stroke attributes in style

stroke stroke color; default is none

stroke-width width of stroke; default is one

stroke-opacity a value between 0.0 (transparent) and 1.0 (opaque, the default)

stroke-dasharray a list of the lengths of dashes and gaps; default is none

Stroke Attributes (cont)

stroke-linecap specifies shape of endpoints: butt (default), round, or square

stroke-linejoin specifies shape of corners: miter (pointed, the default), round, or bevel (flat)

stroke-miterlimit maximum ratio of length of the miter point to width of the lines; default is 4

Fill Attributes

style="attribute: value" specify fill attributes in style

fill fill color; default is black

fill-opacity a value between 0.0 (transparent) and 1.0 (opaque, the default)

fill-rule determines whether a point is inside a shape; nonzero (default) or evenodd

Paths

<path d="command arguments"/>

uppercase commands: absolute coordinates

lowercase commands: relative coordinates

Command	Arguments	Effect
---------	-----------	--------

M m	x y	move to (x, y)
-----	-----	----------------

L l	x y	line to (x, y)
-----	-----	----------------

Z		close path
---	--	------------

H h	x	horizontal line to x
-----	---	----------------------

V v	y	vertical line to y
-----	---	--------------------



By **beccam**
cheatography.com/beccam/

Not published yet.
 Last updated 9th January, 2017.
 Page 2 of 3.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Paths (cont)	Text	
A a rx ry x-axis-rotation large-arc sweep x y	elliptical arc to (x, y); points lie on ellipse with x- radius rx, y-radius ry, rotated x-axis -rotation degrees; if arc < 180°, large- arc is 0; if arc direction is positive, sweep is 1	"a" baseline (x, y) text x="x" y="y"> </text> font-family serif, sans-serif, monospace font-size pt, em, ex, % font-weight bold, normal font-style italic, normal text-decoration none, underline, overline word-spacing +length, normal, -length letter-spacing +length, normal, -length
Q q x1 y1 x y	quadratic Bézier curve to (x, y) using control point (x1, y1)	text-anchor start, middle, end text-align value text-decoration lengthAdjust spacing (def), spacingAndGlyphs writing-mode tb
T t x y	quadratic Bézier curve to (x, y) using reflection of previous T's control pt	glyph-orientation-vertical 0 (letter-spacing: -#), 90 (counter) (c) direction rtl, ltr unicode-bidi bidi-override
C c x1 y1 x2 y2 x y	cubic Bézier curve to (x, y) using control pt 1 (x1, y1) and control pt 2 (x2, y2)	path xlink:href="#path-id">text</textPath> startOffset="val, val%" <tspan style="attributes">spanned text</tspan>
S s x2 y2 x y	cubic Bézier curve to (x, y) using reflection of previous C's control pt for control pt 1 and (x2, y2) for control pt 2	dx="x" or dy="y" offset chars by x or y x="x" or y="y" place chars at x or y rotate="angle" rotate chars by angle baseline-shift super, sub, em, % xml:space="" default, preserve

