SVG Cheat Sheet

Setup

Inside `*.svg` files

```
<svg width="256" height="256" viewBox="0 0 256 256"
     xmlns="http://www.w3.org/2000/svg">
  <!-- SVG code goes in here -->
</svg>
```

The `xmlns` attribute is critical.

width & height denote the dimensions of the SVG graphic—they're very important for browser compatability.

`viewBox` defines a cropping area, following this syntax: `viewBox="x y width height"`

Embedded in HTML files

```
<!doctype html>
<html>
  <head></head>
  <body>
    <svg width="256" height="256" viewBox="0 0 256 256">
      <!-- SVG code goes in here -->
    </svg>
  </body>
</html>
```

The `xmlns` attribute is not necessary.

Shapes

Lines

```
<line x1="0" y1="0" x2="256" y2="256" />
```

`x1`	The line's starting x coordinate
`y1`	The line's starting y coordinate
`x2`	The line's ending x coordinate
`y2`	The line's ending y coordinate

Rectangles

```
<rect x="0" y="0" width="256" height="64" rx="5" ry="5" />
```

`x`	Top left corner's x coordinate
`y`	Top left corner's y coordinate
`width`	Horizontal size of the rectangle
`height`	Vertical size of the rectangle
`rx`	Horizontal radius of the rounding circle
`ry`	Vertical radius of the rounding circle

Circles

```
<circle cx="100" cy="100" r="100" />
```

`cx` Centre x coordinate
`cy` Centre y coordinate
`r` Radius of the circle

Ellipses

```
<ellipse cx="100" cy="100" rx="100" ry="50" />
```

`cx` Centre x coordinate
`cy` Centre y coordinate
`rx` Horizontal radius of the ellipse
`ry` Vertical radius of the ellipse

Polylines

```
<polyline points="0,256 50,150 100,100 150,50" />
```

`points="x1,y1 x2,y2, ..."` coordinates of the corner points

Polygons

Closed polylines

```
<!-- A triangle -->
<polygon points="128,0 256,256 0,256" />
<!-- A hexagon -->
<polygon points="60,20 100,40 100,80 60,100 20,80 20,40" />
```

`points="x1,y1 x2,y2, ..."` coordinates of the corner points

Paths

```
<path d="M100,160 Q128,190 156,160" />
```

`d`	String containing the path commands
`M x y`	Move to the point (x, y)
`m dx dy`	Move from the current point (x, y) to $(x + dx, y + dy)$
`L x y`	Draw a line from the current point to (x, y)
`l dx dy`	Draw a line from the current point (x, y) to $(x + dx, y + dy)$
`H x`	Draw a horizontal line from the current point (x', y') to (x, y')
`h dx`	Draw a horizontal line from the current point (x, y) to $(x + dx, y)$
`V y`	Draw a vertical line from the current point (x', y') to (x', y)
`v dy`	Draw a vertical line from the current point (x, y) to $(x, y + dy)$
`Z` or `z`	Draws a line from the current point of the path to its initial point

`C x1 y1 x2 y2 x y`

Draw a cubic Bézier curve from the current point to the end point (x, y) with start control point (x_1, y_1) and end control point (x_2, y_2)

`c dx1 dy1 dx2 dy2 dx dy`

Draw a cubic Bézier curve from the current point (x, y) to the end point $(x + dx, y + dy)$ with start control point $(x + dx_1, y + dy_1)$ and end control point $(x + dx_2, y + dy_2)$

`S x2 y2 x y`

Draw a smooth cubic Bézier curve from the current point to the end point (x, y) using the end control point (x_2, y_2). The start control point is the reflection of the end control point of the previous curve command about the current point.

`s dx2 dy2 dx dy`

Draw a smooth cubic Bézier curve from the current point (x, y) to the end point $(x + dx, y + dy)$ using the end control point $(x + dx_2, y + dy_2)$. The start control point is the reflection of the end control point of the previous curve command about the current point.

`Q x1 y1 x y`

Draw a quadratic Bézier curve from the current point to the end point (x, y). The control point is (x_1, y_1)

`q dx1 dy1 dx dy`

Draw a quadratic Bézier curve from the current point (x, y) to the end point $(x + dx, y + dy)$. The control point is $(x + dx_1, y + dy_1)$

`T x y`

Draw a smooth quadratic Bézier curve from the current point to the end point (x, y). The control point is the reflection of the control point of the previous curve command about the current point.

`t dx dy`

Draw a smooth quadratic Bézier curve from the current point (x, y) to the end point $(x + dx, y + dy)$. The control point is the reflection of the control point of the previous curve command about the current point.

`A rx ry angle large-arc-flag sweep-flag x y`

Draw an arc curve from the current point to the coordinate (x, y). The center of the ellipse used to draw the arc is determined automatically based on the other parameters of of the two radii r_1 and r_2 of the ellipse, the rotation `angle` in degrees relative to the x-axis. `large-arc=1` choses the large arc and `large-arc=0` choses the small arc. `sweep-arc=1` choses the clockwise turning and `sweep-arc=0` choses the counterclockwise turning.

`a rx ry angle large-arc-flag sweep-flag dx dy`

Draw an arc curve from the current (x, y) point to a point $(x + dx, y + dy)$ The center of the ellipse used to draw the arc is determined automatically based on the other parameters (see above).

Styling shapes and strokes

Fill

```
<circle fill="orange" cx="100" cy="100" r="100" />
```

`fill` Sets the fill color of a shape.

keywords: `#F9AC03`, `rgb(0,128,0)` `rgba(0,128,0, 0.5)`

Stroke

```
<circle stroke="orange" cx="100" cy="100" r="100" />
```

`stroke` Sets the color of a stroke

keywords: `#F9AC03`, `rgb(0,128,0)` `rgba(0,128,0, 0.5)`

Stroke width

```
<line stroke-width="10" x1="0" y1="0" x2="100" y2="200" />
```

`stroke-width` Sets the thickness of the stroke.