Working with SVGs

Michelle Seeto

SVG

=

Scalable Vector Graphics

Raster Graphics

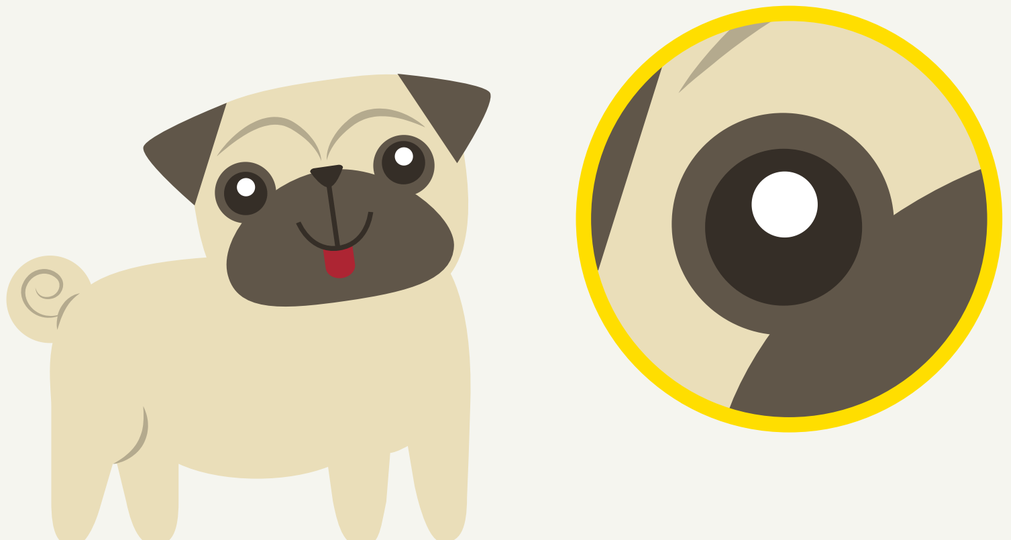
Composed of pixels on a grid, where each pixel is assigned a colour value

Resolution-dependent – cannot be enlarged without degrading their quality

Vector Graphics

Small graphics that use **math** to display images

Can be enlarged **without losing quality**



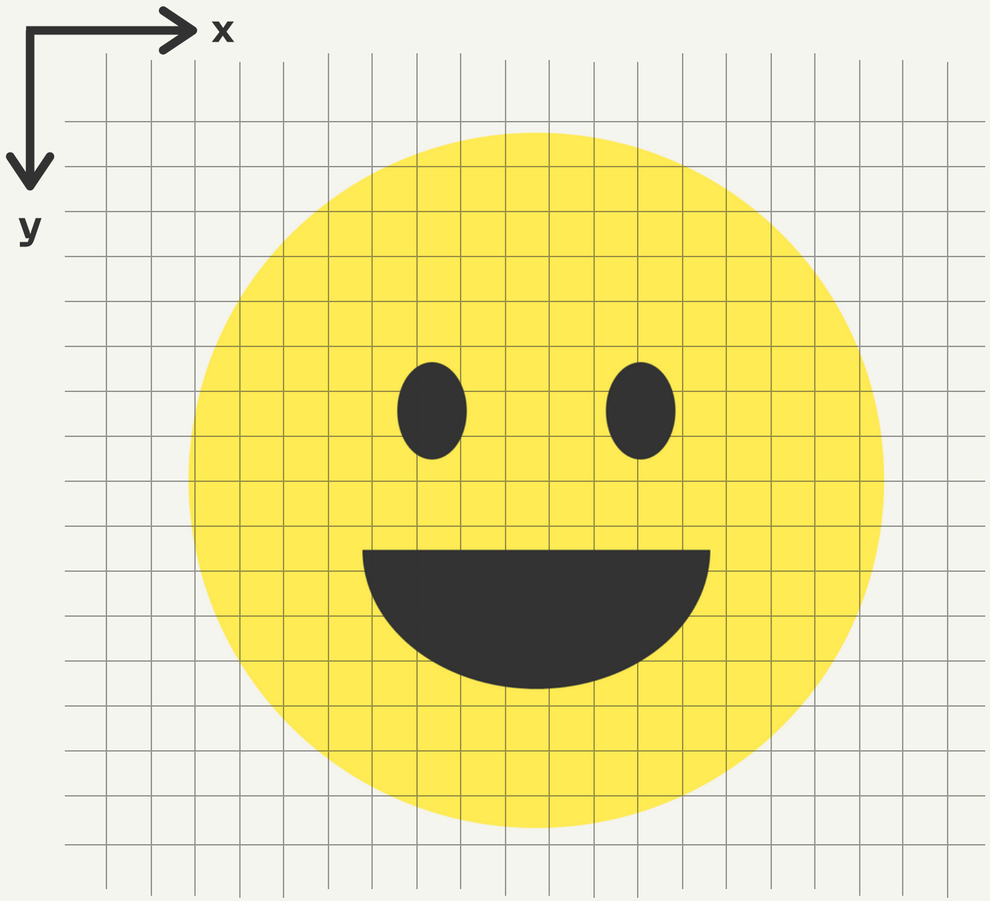
What's in an SVG?

A **vector** is a geometric object with **magnitude** and **direction**.

```
<svg></svg>
```

What's in an SVG?

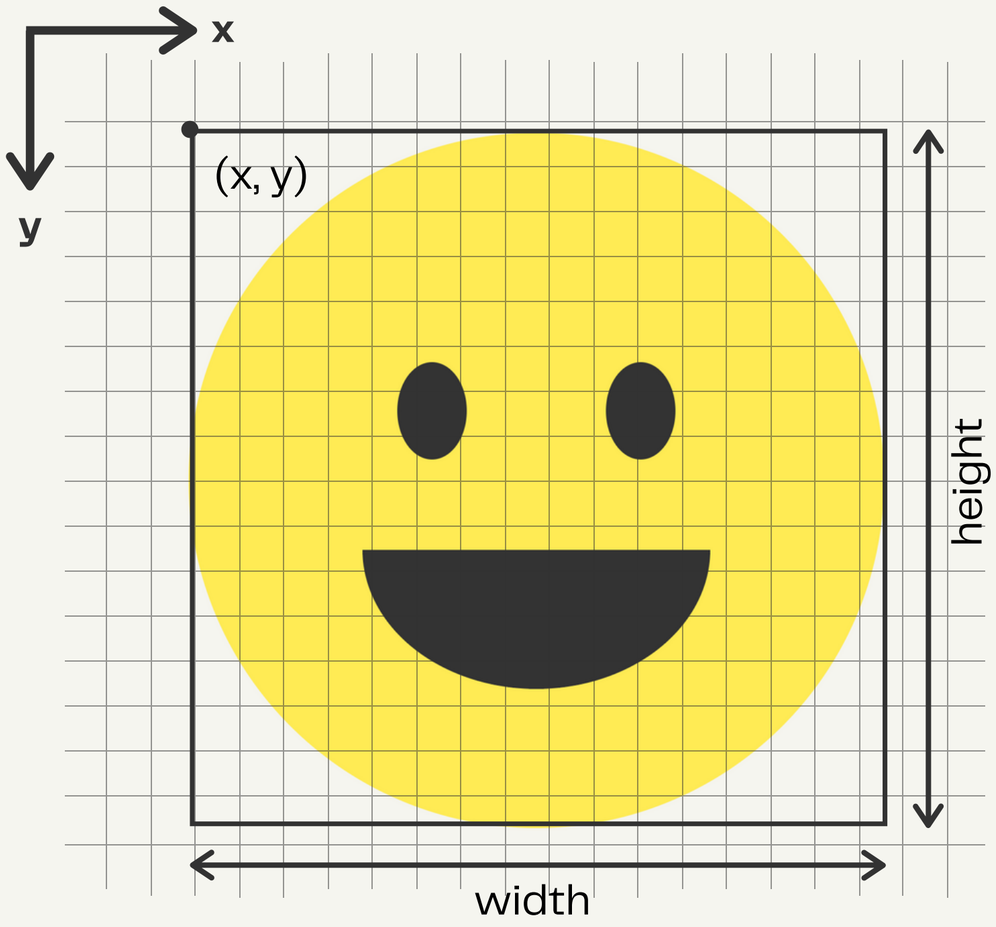
A **vector** is a geometric object with **magnitude** and **direction**.




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

- **Top-left coordinate system**
  - Coordinates expressed as x, y

What's in an SVG?

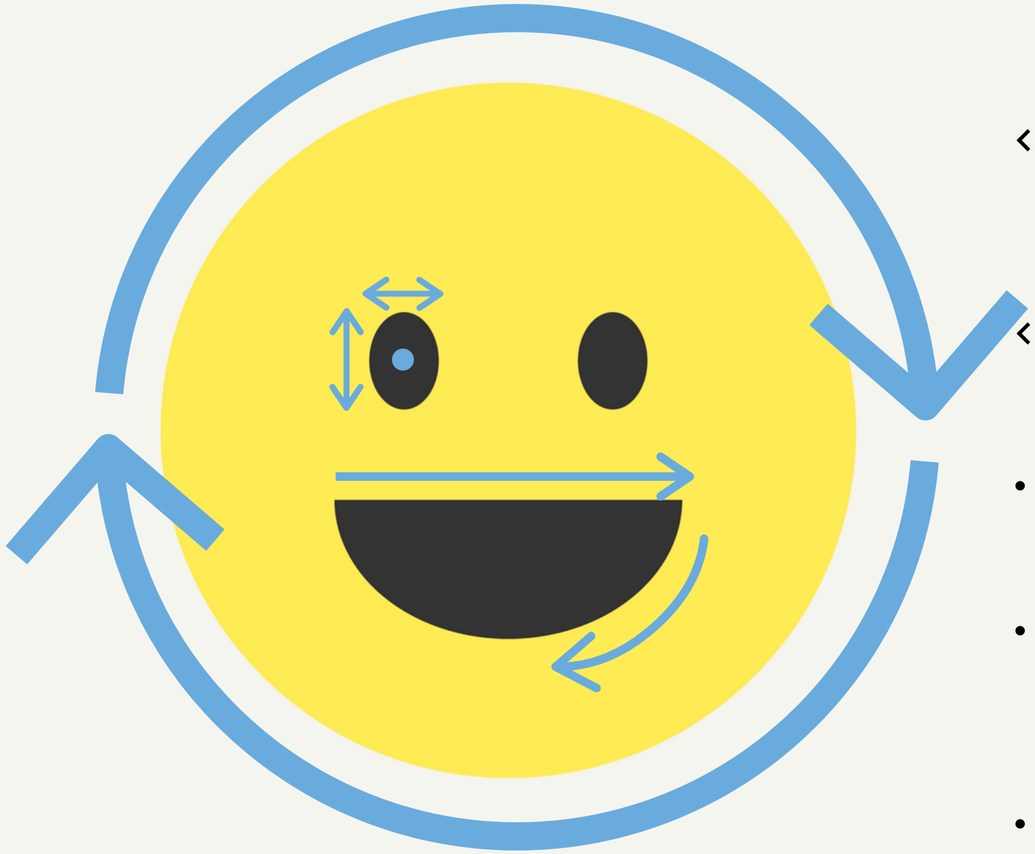



```
<svg viewBox="0 0 100 100" width="100" height="100">
</svg>
```

Note: not the same coordinate system!

- Top-left coordinate system
  - Coordinates expressed as x, y
- **Viewbox – the visible region of the SVG**
  - Think of it as a window looking out to the scene
  - Anything outside the viewport is clipped from view

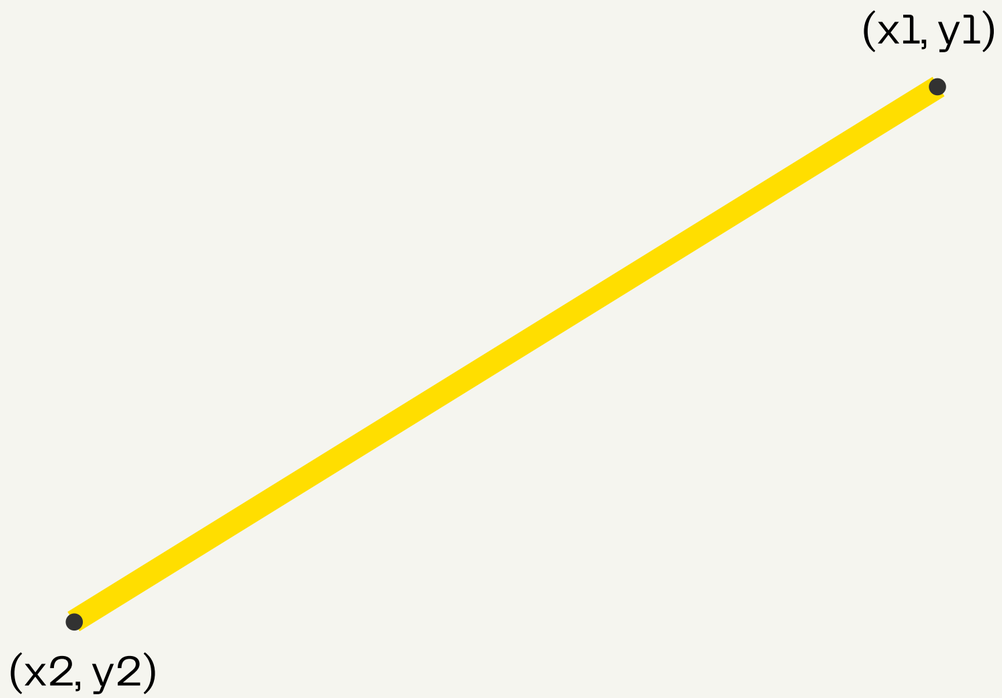
What's in an SVG?



```
<svg viewBox="0 0 100 100" width="100" height="100">
    <path ... />
    <circle ... />
    <line ... />
</svg>
```

- Top-left coordinate system
  - Coordinates expressed as x, y
- Viewbox – the visible region of the SVG
  - Think of it as a window looking out to the scene
  - Anything outside the viewport is clipped from view
- **Collection of paths and vector shapes**, which define the SVG's contents

SVG shapes: line

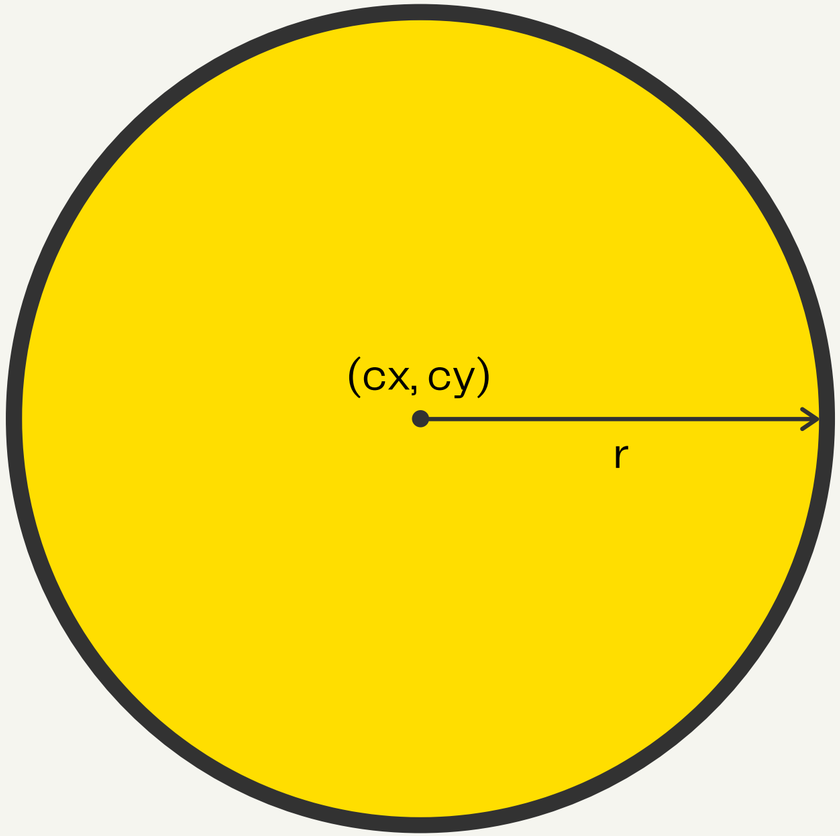



```
<svg viewBox="0 0 100 100" width="100" height="100">
   <line x1="80" y1="10" x2="20" y2="70" stroke="yellow"/>
</svg>
```

Common styling:

- `stroke` for line colour (must be specified)
- `stroke-width` for line thickness
- `stroke-linecap` to specify whether the endpoint is round, square etc.

SVG shapes: circle

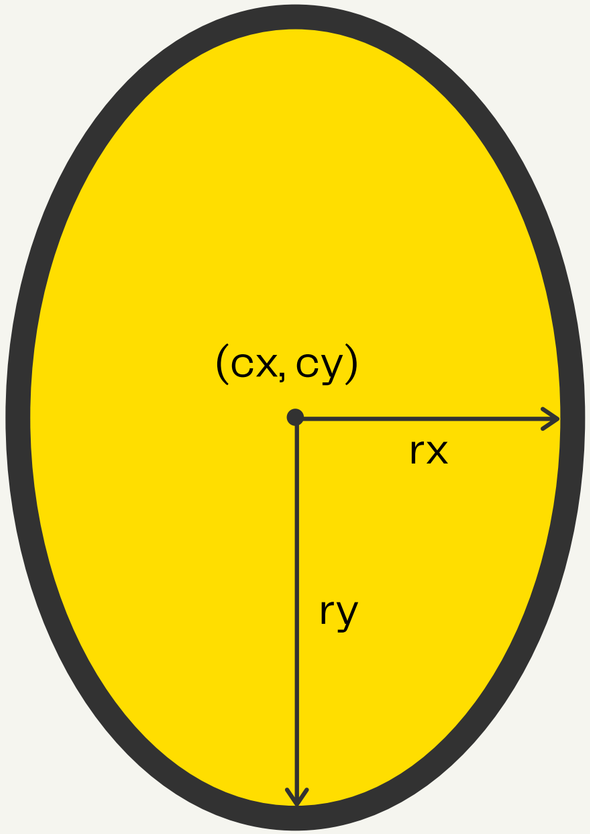



```
<svg viewBox="0 0 100 100" width="100" height="100">
   <circle cx="50" cy="50" r="50" />
</svg>
```

Common styling:

- `fill` for filled-in colour
- `stroke` for outline colour
- `stroke-width` for line thickness

SVG shapes: ellipse

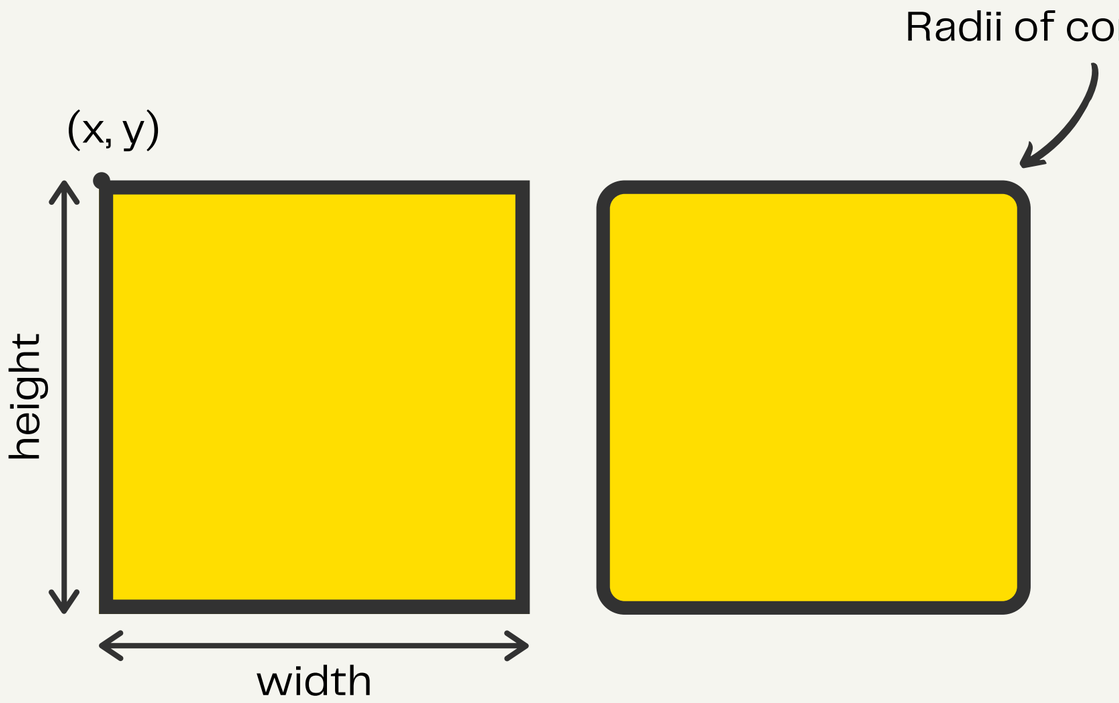



```
<svg viewBox="0 0 100 100" width="100" height="100">
   <ellipse cx="50" cy="50" rx="50" ry="25" />
</svg>
```

Common styling:

- `fill` for filled-in colour
- `stroke` for outline colour
- `stroke-width` for line thickness

SVG shapes: rect



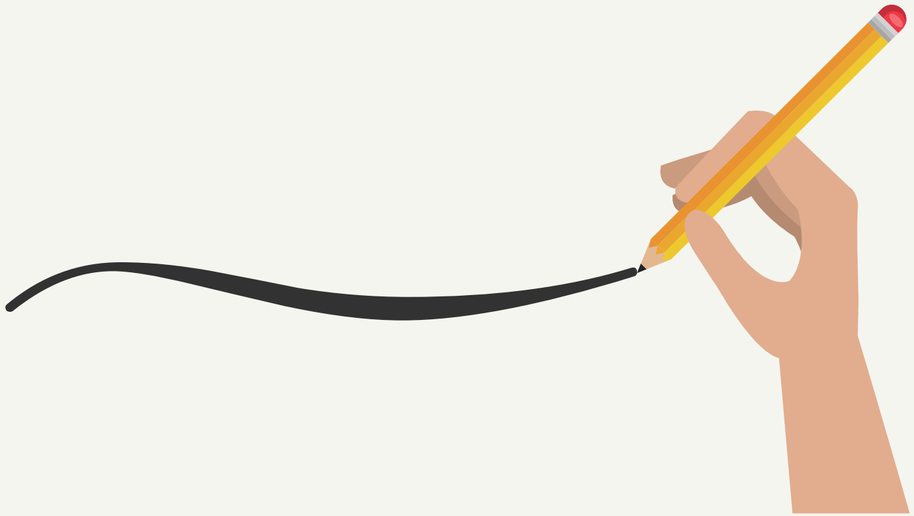

```
<svg viewBox="0 0 100 100" width="100" height="100">
   <rect x="0" y="0" width="50" height="50" rx="5" ry="5"/>
</svg>
```

Common styling:

- `fill` for filled-in colour
- `stroke` for outline colour
- `stroke-width` for line thickness

SVG path

Any generic shape can be defined by an SVG path. This is where the geometry really kicks in.
Think of it as a set of commands that could be given to someone with a pencil and paper.
Commands are written in terms of absolute or relative coordinates.



```
<svg viewBox="0 0 100 100" width="100" height="100">
   <path d="M0 0 L100 100 L0 100 Z" />
</svg>
```

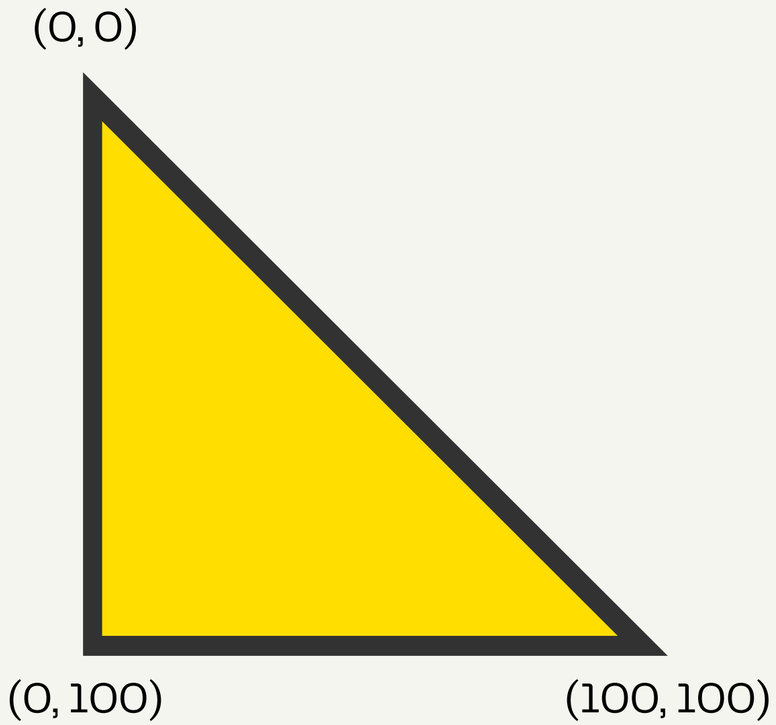
Common styling:

- `fill` for filled-in colour
- `stroke` for outline colour
- `stroke-width` for line thickness

SVG path commands

	Absolute commands:	**Relative commands:**
M = move to	**M** x y	**m** dx dy
L = line to	**L** x y	**l** dx dy
H = horizontal line to	**H** x	**h** dx
V = vertical line to	**V** y	**v** dy
C = curve to	**C** x1 y1, x2 y2, x y	**c** dx1 dy1, dx2 dy2, dx dy
S = smooth curve to	**S** x2 y2, x y	**s** dx2 dy2, dx dy
Q = quadratic Bézier curve to	**Q** x1 y1, x y	**q** dx1 dy1, dx dy
T = smooth quadratic Bézier curve to	**T** x y	**t** dx dy
A = elliptical arc	**A** rx ry x-axis-rotation large-arc-flag sweep-flag x y	**a** rx ry x-axis-rotation large-arc-flag sweep-flag dx dy
Z = close path (straight line from the current position to the first point in the path)	**Z**	**z**

SVG path




```
<svg viewBox="0 0 100 100" width="100" height="100">
   <path d="M0 0 L100 100 L0 100 Z" />
</svg>
```

Commands:

1. Move to (0, 0)
2. Line to (100, 100)
3. Line to (0, 100)
4. Close path

or

```
<svg viewBox="0 0 100 100" width="100" height="100">
   <path d="M0 0 l100 100 l-100 0 z" />
</svg>
```

Commands:

1. Move to (0, 0)
2. Line in direction +100 units in x-axis, +100 units in y-axis
3. Line in direction -100 units in x-axis, no change in y-axis
4. Close path

Bringing it all together

Thank you!