

# Intro to SVG

Let's start →





# Estee Tey

- 🖥️ Grad Software Developer at Thoughtworks
- ✎️ Writes about Web Dev, UI, Dev growth
- 🎨 Experienced in creating mockups & graphics

🐙 [lyqht](#)    🐦 [estee\\_tey](#)

# Expectations




## No Prerequisites

`expect(target_audience).toBeAny()`

## Learn something

`expect().toIncrease();`

## Your time to be worth it

`expect().toYield()`

## Have fun thinking

`expect().andThen()`

# Table of Contents

1. What is SVG?
  2. Why do we use SVG?
  3. How do we get SVG?
  4. How to use SVG?
- 

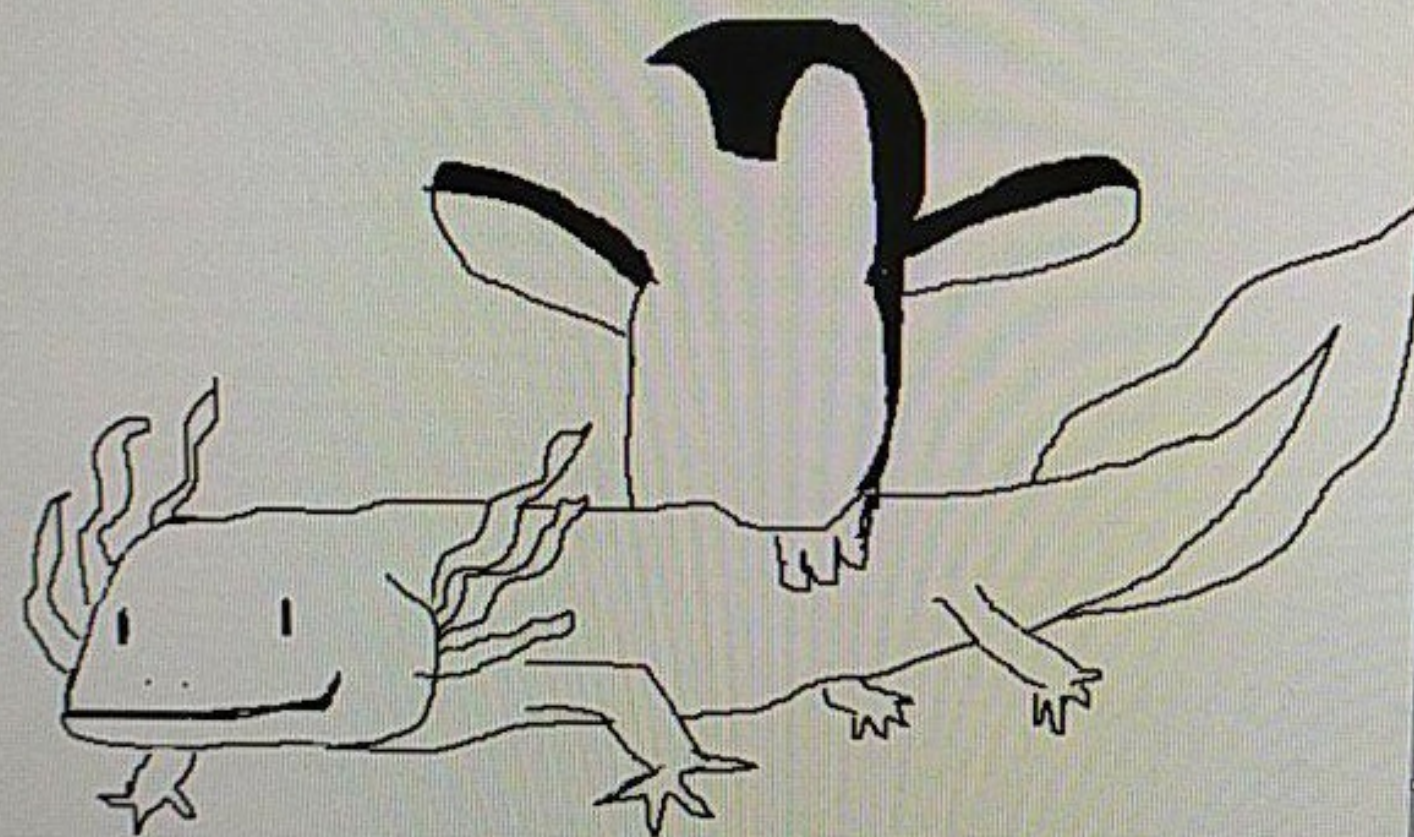
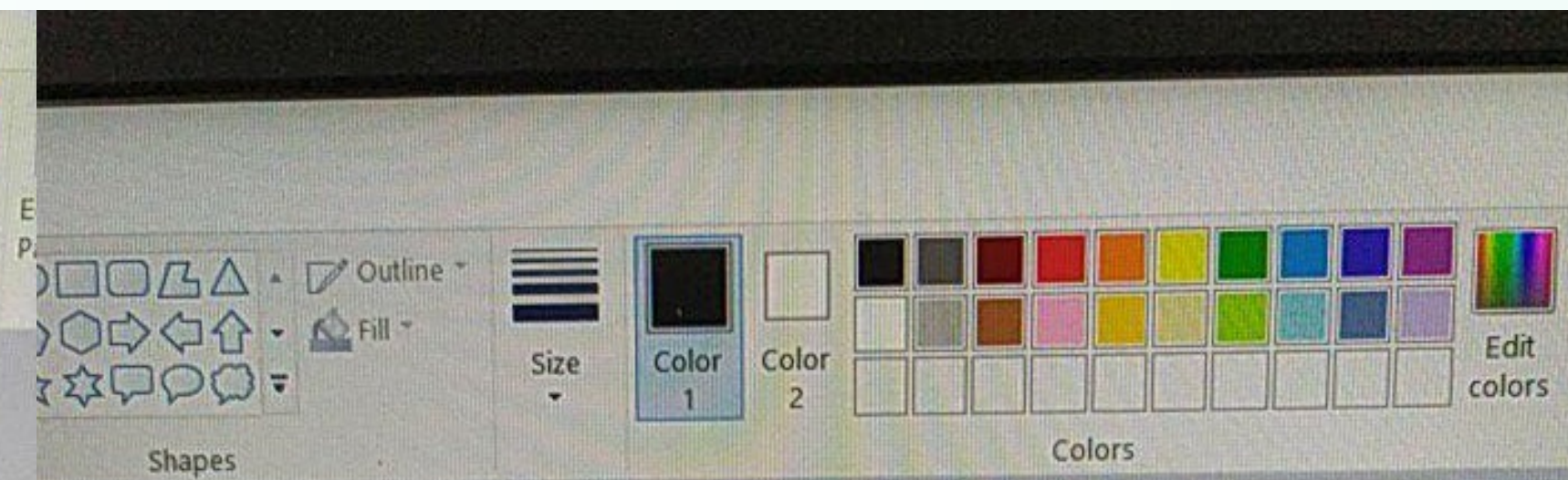
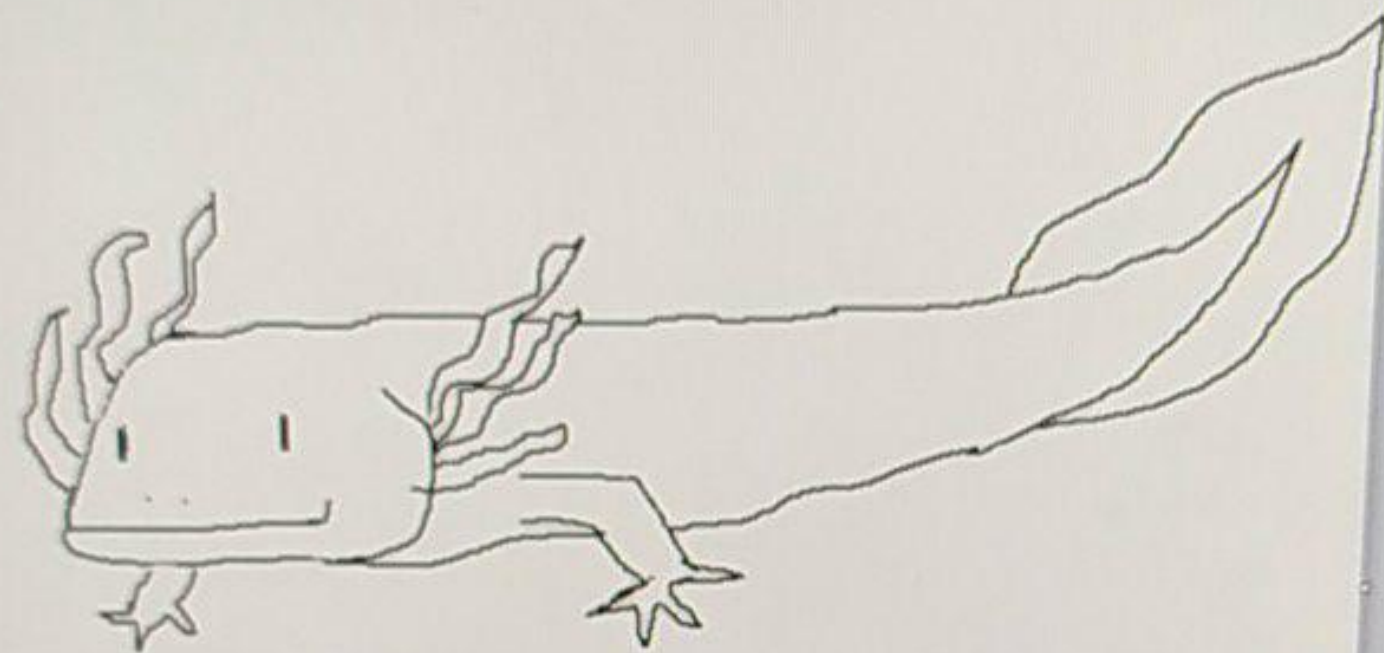
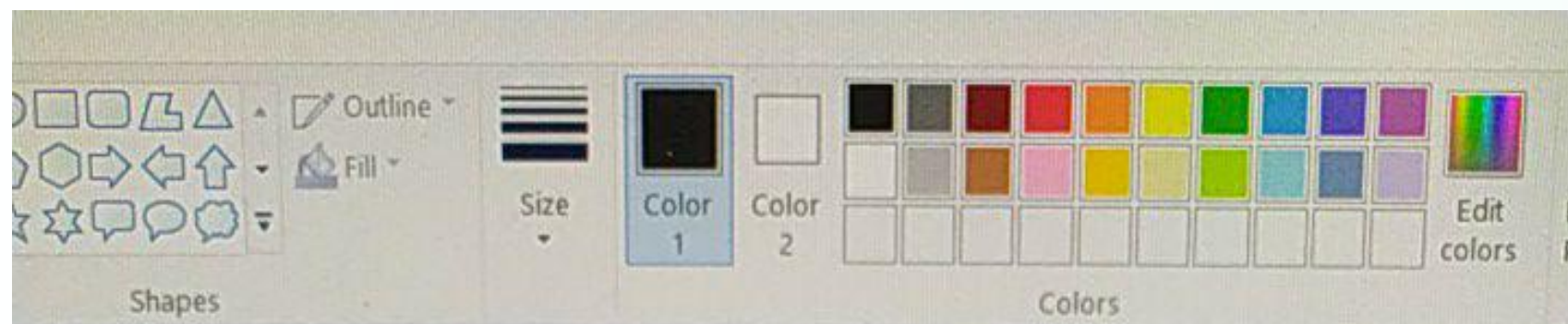
What is SVG?

# Images

Story Time! →



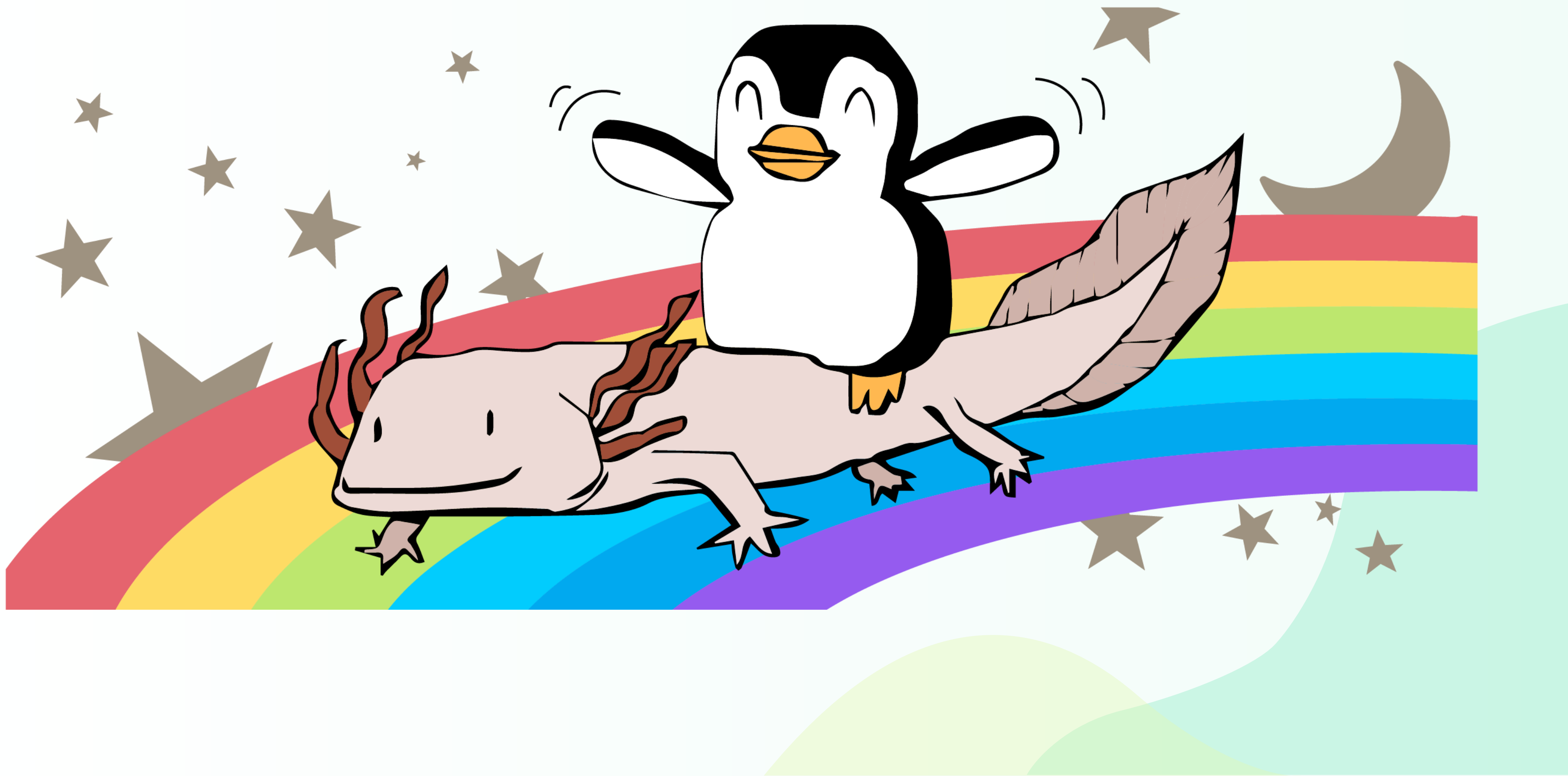












# Raster vs Vector Images

painting with pixels



drawing with vectors



vector-conversions.com

# Image File Types

File Extension	Description
.gif	Graphics Interchange Format
.png	Portable Network Graphic
.jpg / .jpeg	Image format by Joint Photographic Experts Group
.webp	A superior image format of .png & .jpg
.svg	Scalable Vector Graphics



# Image File Types

File Extension	Color Modes	Compression	Usage
.gif	Indexed Color	Loseless	Animated images
.png	Greyscale, True Color, Alpha	Loseless (better than .GIF)	Static Line art, iconic graphics where transparency matters.
.jpg / .jpeg	True Color	Lossy	Photographs, realistic images of people, venues etc.
.webp	Depends on compression	Loseless / Lossy	A superior image format of .png & .jpg
.svg	Anything that can be specified using CSS color syntax.	NA	UI that requires to be redrawn accurately at different sizes

# How to get SVG

1. Create SVG from scratch
2. Generate SVG
3. Extract from websites

# Create SVG with 3 basic elements

```
<rectangle />
```

```
<circle />
```

```
<polygon />
```

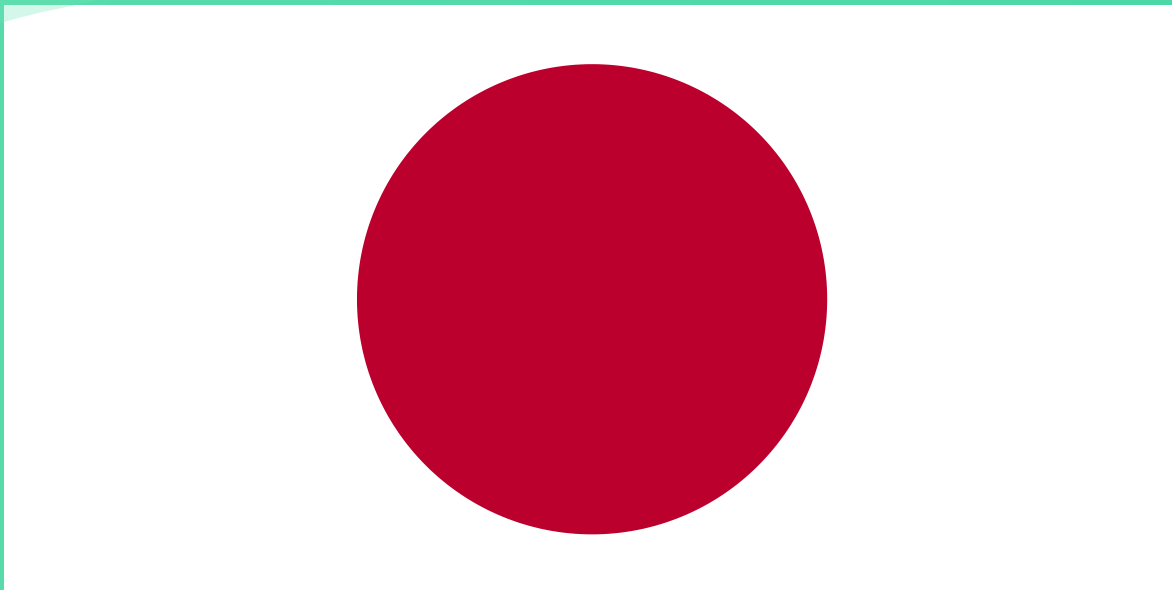
Code Time! →

# Create SVG with 3 simple shapes in code





# JP flag in SVG



```
1  <svg>
2    <rect width="100%" height="100%" fill="white" />
3    <circle cx="50%" cy="50%" r=60 fill="#BC002D" />
4  </svg>
```





# TH flag in SVG



```
1  <svg>
2    <rect width="100%" height="20%" y="0" fill="red" />
3    <rect width="100%" height="20%" y="20%" fill="white" />
4    <rect width="100%" height="20%" y="40%" fill="blue" />
5    <rect width="100%" height="20%" y="60%" fill="white" />
6    <rect width="100%" height="20%" y="80%" fill="red" />
7  </svg>
```



# VN flag in SVG



```
1  <svg>
2    <defs>
3      <polygon id="star" fill="yellow"
4        points="100,10 40,198 190,78 10,78 160,198" />
5    </defs>
6    <rect width="100%" height="100%" fill="red" />
7    <svg viewBox="100 0 600 600" x="25%" y="25%">
8      <use href="#star">
9      </use>
10   </svg>
11 </svg>
```



# Polygon vs Polyline



```
1  <polygon points="100,10 40,198 190,78 1  
0,78 160,198" fill='#abcbca' stroke='#123123'  
stroke-width='5' />
```



```
1  <polyline points="100,10 40,198 190,78 1  
0,78 160,198" fill='#abcbca' stroke='#123123'  
' stroke-width='5' />
```



# SG flag in SVG



```
1  <svg>
2    <rect width="100%" height="50%" fill="red" />
3    <rect width="100%" height="50%" y="50%" fill="white" />
4    <circle cx='15%' cy='25%' r='30' fill='white' />
5    <circle cx='20%' cy='25%' r='30' fill='red' />
6    <svg viewBox="600 -100 1000 1800">
7      <use href="#white-star" />
8      <use href="#white-star" x="-20%" y="10%" />
9      <use href="#white-star" x="20%" y="10%" />
10     <use href="#white-star" x="-12.5%" y="22%" />
11     <use href="#white-star" x="12.5%" y="22%" />
12   </svg>
13 </svg>
```

# Create SVG with Design Tools





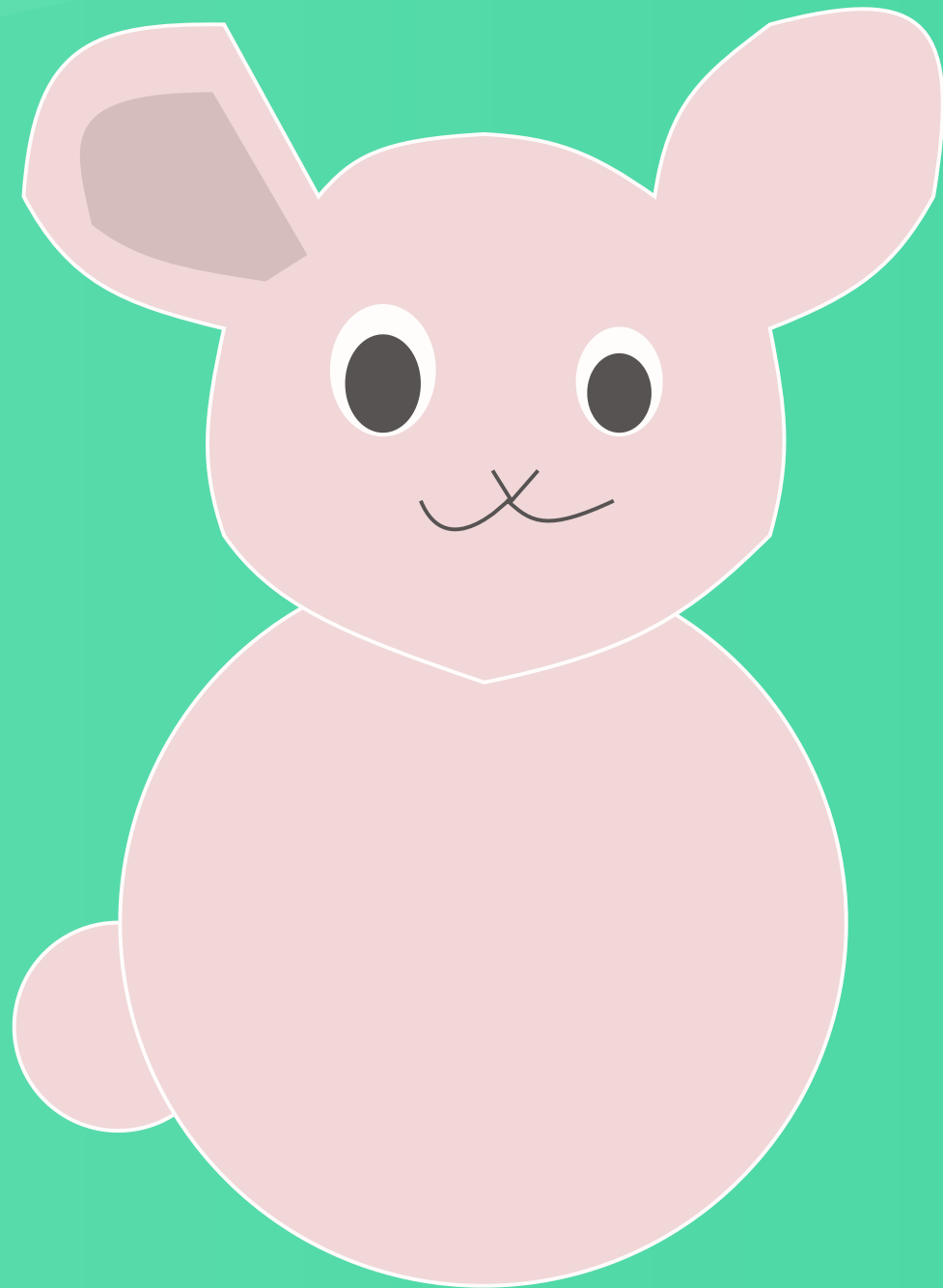
# Figma Demo

Ellipse, Polygon, Paths, Bezier Curves





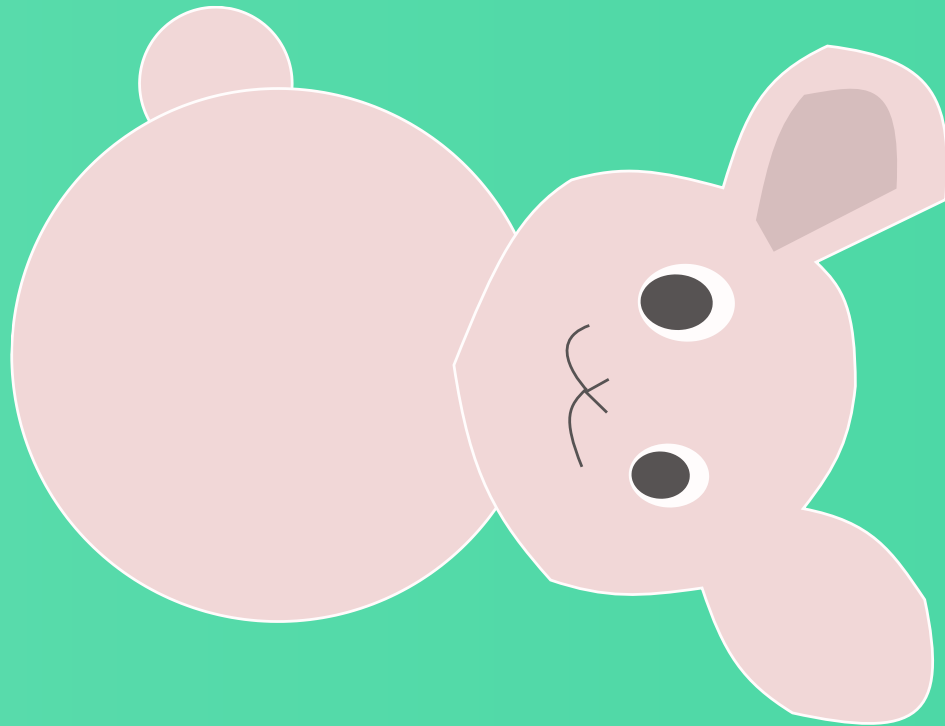
# Setting SVG File as an image source



```
1 
```



# Setting SVG File as an image source

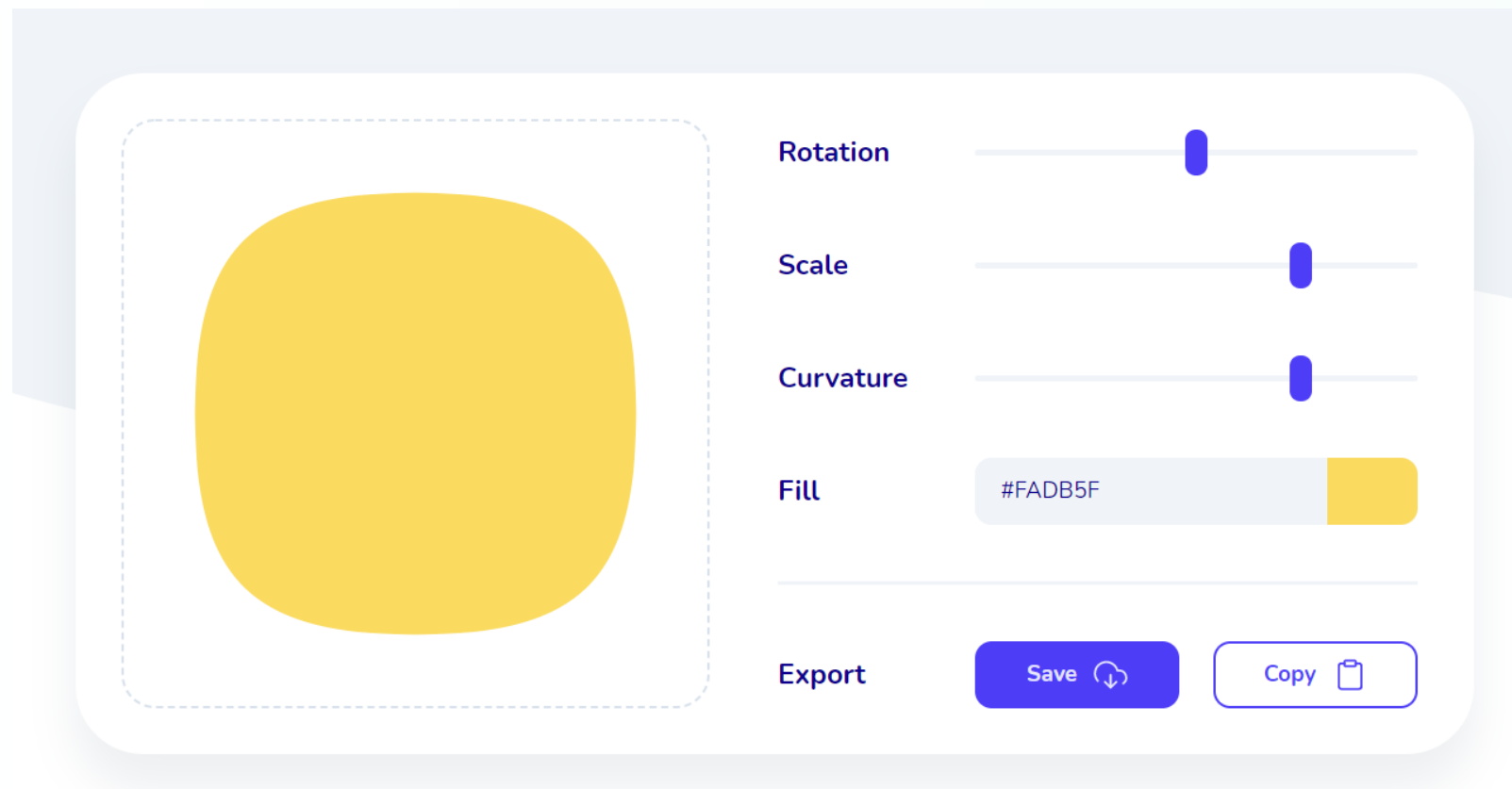


```
1 
```

```
1 .special-image {
2   height:240px;
3   animation:spin 4s linear infinite;
4 }
5
6 @keyframes spin {
7   100% {
8     transform:rotate(360deg);
9   }
10 }
```

# Generate SVG

## 1. Squicley for generating Squircles



# Generate SVG

2. [SVGBackgrounds.com](https://svgbackgrounds.com) for generating backgrounds





# Extract from websites

Demo with Hacktoberfest's SVG

⌈ P.S. If you use any extracted images for your own websites or apps, please remember to give **attribution**.

# Bonus: Optimization of SVG

SVGOMG

# Bonus for React Devs: convert from SVG to JSX

SVG to JSX Demo

# Summary

1. What is SVG?
2. Why do we use SVG?
3. How do we get SVG?
4. How to use SVG?


# SVG Element Cheatsheet

	Shape	Usage Example
1.	Square	<code>&lt;rect width="40" height="40" /&gt;</code>
2.	Rectangle	<code>&lt;rect width="80" height="40" /&gt;</code>
3.	Circle	<code>&lt;circle cx="50%" cy="50%" r="60" /&gt;</code>
4.	Ellipse	<code>&lt;ellipse cx="100" cy="50" rx="80" ry="40" /&gt;</code>
5.	Polygon	<code>&lt;polygon points="100,10 40,198 190,78 10,78 160,198" /&gt;</code>
6.	Polyline	<code>&lt;polyline points="100,10 40,198 190,78 10,78 160,198" /&gt;</code>

# Topics for you to explore more

- More complex SVG elements such as patterns, filters, paths
  - Accessibility of SVGs
  - Performance of SVGs
- 

# More Resources

- [MDN Web Docs on SVG](#)
  - [CSS Tricks - how to scale SVG](#)
  - [Smashing Magazine - SVG Generators](#)
  - [Frontend Masters - SVG Essentials & Animations v2](#)
- 



# Thank you!

Hope you enjoyed the L&L 😊

Any questions?

–

Feedback form

