

SVG Cheat Sheet

Shapes

Rectangle

```
<rect x="10" y="10" width="80" height="50" fill="skyblue">
```

Circle

```
<circle cx="60" cy="60" r="40" fill="orange">
```

Ellipse

```
<ellipse cx="80" cy="50" rx="60" ry="30" fill="pink">
```

Line

```
<line x1="10" y1="10" x2="90" y2="90" stroke="black" stroke-width="2" />
```

Polygon (Closed Shape)

```
<polygon points="50,5 100,100 0,100" fill="limegreen">
```

Polyline (Open Shape)

```
<polyline points="0,40 40,40 40,80 80,80" fill="none" stroke="blue" stroke-width="2" />
```

SVG provides several basic shapes for building graphics: rectangles, circles, ellipses, lines, polygons, and polylines. These elements are easy to use and customize with color, stroke, and dimensions.

Text & Fonts

Basic Text

```
<text x="10" y="40" font-size="20">Hello SVG</text>
```

Centered Text

```
<text x="50" y="50" text-anchor="middle" font-size="16">Centered</text>
```

Multi-line with

```
<text x="10" y="20" font-size="14">
  Line 1
  <span x="10" dy="20">Line 2</span>
</text>
```

You can add editable text with the `<text>` element. Position it with **x** and **y**, align it using **text-anchor**, and style it using font properties or child **** tags.

Colors & Fills

Fill and Stroke

```
<circle cx="50" cy="50" r="40" fill="gold" stroke="black" stroke-width="2">
```

Linear Gradient

```
<defs>
  <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
    <stop offset="0%" stop-color="red">
    <stop offset="100%" stop-color="yellow">
  </linearGradient>
</defs>
<rect x="10" y="10" width="100" height="100" fill="url(#grad1)">
```

Pattern Fill

```
<defs>
  <pattern id="dots" width="10" height="10" patternUnits="userSpaceOnUse">
    <circle cx="5" cy="5" r="2" fill="black">
  </pattern>
</defs>
<rect width="100" height="100" fill="url(#dots)">
```

Use **fill** and **stroke** to color your shapes, or apply gradients and patterns by referencing **url(#id)**. Gradients allow smooth transitions between colors, while patterns tile repeating elements.

Animation

Animate Position

Basic Structure

Basic SVG Element with Width & Height

```
<svg width="200" height="200" xmlns="http://www.w3.org/2000/svg">
  <rect x="10" y="10" width="50" height="50" fill="red">
</svg>
```

SVG with ViewBox (Scalable)

```
<svg viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg">
  <circle cx="50" cy="50" r="40" fill="blue">
</svg>
```

Responsive SVG with preserveAspectRatio

```
<svg width="100%" height="100%" viewBox="0 0 500 500" preserveAspectRatio="xMidYMid">
  <text x="50" y="50" font-size="24">Hello SVG</text>
</svg>
```

This is the foundation of every SVG file. The `<svg>` element defines the canvas, and attributes like **width**, **height**, and **viewBox** control its size and scaling behavior.

Path Commands

Rectangle Using Path

```
<path d="M10 10 H 90 V 90 H 10 Z" fill="lightgreen">
```

Cubic Bezier Curve

```
<path d="M50 10 C 70 10, 70 40, 50 40" stroke="red" fill="none">
```

Quadratic Bezier Curve

```
<path d="M20 80 Q 50 10, 80 80" stroke="blue" fill="none">
```

Path Command Reference

- M** - move to
- L** - line to
- H/V** - horizontal/vertical line
- C** - cubic Bezier curve
- S** - smooth cubic curve
- Q** - quadratic Bezier curve
- T** - smooth quadratic curve
- A** - arc
- Z** - close path

The `<path>` element lets you draw custom shapes using a series of commands and coordinates. It's one of the most powerful **SVG tools**, allowing you to create curves, arcs, and complex figures.

Transforms

Translate (Move)

```
<rect x="0" y="0" width="50" height="50" fill="blue" transform="translate(20,20)">
```

Rotate

```
<rect x="50" y="20" width="60" height="30" fill="green" transform="rotate(45)">
```

Scale

```
<circle cx="50" cy="50" r="20" fill="orange" transform="scale(1.5)">
```

Multiple Transforms

```
<rect width="50" height="50" fill="purple" transform="translate(20,20) rotate(45) scale(1.5)">
```

Use the **transform** attribute to translate (move), rotate, or scale elements. Transformations are applied in the order they appear. Coordinates in rotation use the syntax **rotate(angle cx cy)**.

Filters

Gaussian Blur

```
<defs>
  <filter id="blur">
    <feGaussianBlur in="SourceGraphic" stdDeviation="4">
  </filter>
</defs>
<circle cx="60" cy="60" r="40" fill="blue" filter="url(#blur)">
```

```
<circle cx="20" cy="50" r="10" fill="tomato">
  <animate attributeName="cx" from="20" to="120" dur="2s" repeatCount="infinite">
</circle>
```

🌈 Animate Fill Color

```
<rect x="20" y="20" width="60" height="60">
  <animate attributeName="fill" values="red;blue;green" dur="3s" repeatCount="infinite">
</rect>
```

🔄 Animate Rotation

```
<rect x="50" y="50" width="40" height="40" fill="purple">
  <animateTransform attributeName="transform" type="rotate" from="0 70 70" to="360 70 70">
</rect>
```

SVG has built-in animation support using **<animate>** and **<animateTransform>**. You can animate any attribute, such as position, color, or rotation, over time.

Interactivity 🖱️

🖱️ Inline Click Event

```
<circle cx="60" cy="60" r="40" fill="lightblue" onclick="alert('You clicked!')">
```

💡 JavaScript Event Listener

```
<svg id="htmlcheatsheet" width="100" height="100">
  <rect id="box" x="10" y="10" width="80" height="80" fill="tomato">
</svg>
<script>
  document.getElementById("box").addEventListener("click", () => {
    alert("Box clicked!");
  });
</script>
```

🔍 CSS Hover Effect

```
<style>
  circle:hover {
    fill: gold;
    cursor: pointer;
  }
</style>
<circle cx="50" cy="50" r="30" fill="gray">
SVG supports interactivity through HTML-style events like onclick or by attaching listeners in JavaScript. You can also use CSS for hover effects and transitions.
```

Embedding & Responsiveness 🌐

🌟 Inline SVG

```
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" fill="red">
</svg>
```

🖼️ SVG as Image

```

```

🎨 CSS Background

```
div {
  background-image: url('icon.svg');
  width: 100px;
  height: 100px;
}
```

📱 Responsive SVG

```
<svg viewBox="0 0 100 100" width="100%" height="auto">
  <rect width="100" height="100" fill="lightblue">
</svg>
```

You can embed SVG inline in HTML, as an image, or as a CSS background. Use **viewBox** and percentage-based sizing to make SVGs responsive across screen sizes.

🌑 Drop Shadow

```
<defs>
  <filter id="shadow">
    <feDropShadow dx="4" dy="4" stdDeviation="3" flood-color="gray">
    </filter>
</defs>
<rect x="20" y="20" width="80" height="60" fill="orange" filter="url(#shadow)">
```

🎨 Color Matrix (Grayscale)

```
<defs>
  <filter id="grayscale">
    <feColorMatrix type="matrix" values="0.33 0.33 0.33 0 0
      0.33 0.33 0.33 0 0
      0.33 0.33 0.33 0 0
      0 0 0 1 0">
    </filter>
</defs>
<image href="photo.jpg" x="0" y="0" width="200" filter="url(#grayscale)">
```

Filters let you apply visual effects like blur, shadow, and color transformations. Define a filter in **<defs>**, then apply it to any shape using **filter="url(#id)"**.

Use & Reuse ♻️

🏠 Define a Symbol

```
<svg style="display:none;">
  <symbol id="star" viewBox="0 0 100 100">
    <polygon points="50,5 61,35 95,35 68,57
      78,91 50,70 22,91 32,57
      5,35 39,35" fill="gold">
    </symbol>
</svg>
```

🔗 Reuse with <use>

```
<svg width="100" height="100">
  <use href="#star" x="0" y="0">
  <use href="#star" x="110" y="0" fill="blue">
</svg>
Use <symbol> inside a hidden SVG to define reusable elements, then display them with <use href="#id>. You can position and style each instance independently.
```

Accessibility ♿

📄 Title and Description

```
<svg role="img" aria-labelledby="title desc" width="100" height="100">
  <title id="title">Warning Icon</title>
  <desc id="desc">A triangle with an exclamation mark</desc>
  <polygon points="50,5 95,95 5,95" fill="yellow" stroke="black">
  <text x="50" y="75" font-size="40" text-anchor="middle" fill="black">!</text>
</svg>
```

👤 Hide from Assistive Tech

```
<svg aria-hidden="true" focusable="false">
  <circle cx="50" cy="50" r="40" fill="gray">
</svg>
Use <title> and <desc> for screen reader context. The aria-labelledby attribute connects them. To hide decorative SVGs, use aria-hidden="true" and focusable="false".
```

HTML Cheat Sheet 📋

- 🌐 [htmlcheatsheet.com](#) Visit HTML Cheat Sheet.com
- 🛠️ for the interactive SVG editor and code optimizer