Engineer

SVG

an XML language for drawing graphics that scale

What it is

SVG, short for Scalable Vector Graphics, is a language for describing two-dimensional graphics as shapes rather than as a grid of pixels. The file says what to draw: a circle here, a line from this point to that one, this outline filled in blue. The browser works out the pixels at whatever size the drawing is shown, and that is where the scaling comes from: there are no pixels in the file to stretch.

The W3C defines it. Its latest version, SVG 2, is a Candidate Recommendation Snapshot dated 6 October 2026, expected to become a Recommendation no earlier than 1 December 2026, and it builds on SVG 1.1 Second Edition. As the specification sums itself up, SVG is built on XML, can combine vector shapes with raster images, and takes styles, script and animation. A drawing can stand alone, sit inside an HTML page or be embedded in another XML language.

The vocabulary is small. Six basic shapes, `rect`, `circle`, `ellipse`, `line`, `polyline` and `polygon`, cover the simple cases, and a `path` draws anything else from a short string of commands. `text` places real, selectable text, a group (`g`) lets several shapes share a style or a transform, and a gradient or a pattern can paint a shape in place of a single colour. Filter effects, a chapter of SVG 1.1, now have a specification of their own. The initial coordinate system has its origin at the top left, x growing to the right and y growing downwards.

The specification defines SVG in terms of a document object model rather than one particular file format. A drawing can be a file of its own, served as `image/svg+xml` with the extension `.svg`, or it can be an `svg` element written straight into an HTML page. A standalone file must declare the SVG namespace; in HTML the parser puts the elements in the right namespace by itself, so no `xmlns` attribute is needed.

SVG is not a raster format like PNG or JPEG, which store a colour for every pixel and go soft when enlarged. A photograph belongs in one of those. SVG suits what was drawn: icons, logos, diagrams, charts, maps and lettering. It can also carry a raster image inside it, which is what the specification means by mixed graphics.

Use the concept in a decision

Put “What is SVG?” to work in a real decision. We can help relate the concept to the system you are building or buying. Interfaces that load quickly, work on a phone and behave the same in every browser your customers happen to use.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Directed full-stack GIS map development — full-stack mapping, from database to browser

Directed full-stack GIS map development Get in touch →

Why it matters

One file serves every screen. An icon drawn once is sharp on an old monitor, on a phone with several device pixels to every CSS pixel and on paper, with no separate exports at each size to keep in step. Simple artwork also makes a small file, because it describes a few outlines instead of storing thousands of pixels, and the specification registers `.svgz` for an SVG file compressed with gzip.

Because the markup is text, it fits the way a team already works. A designer’s export can be read in a code review and compared line by line in version control, and a script can generate a drawing from data, which is how many charts are made. Written inline in a page, the drawing’s elements are part of the DOM: CSS styles them, a script can change them, and they follow hover, focus and the reader’s colour scheme like everything else on the page.

It also carries meaning that a bitmap cannot. Text in an SVG stays text, and in a processing mode that allows interaction the specification requires a browser to let the reader search for it and select it. A `title` element gives a drawing, or one part of it, an accessible name, and SVG 2 gives the `svg` element the implicit ARIA role `graphics-document`. A chart drawn in SVG can have labelled parts that a screen reader reaches one by one; the same chart exported as a PNG is a single picture with a single text alternative.

What it looks like in practice

Here is a small arrow in a rounded frame, written inline in an HTML page:

```
<svg viewBox="0 0 120 40" width="240" height="80">
  <title>Next step</title>
  <g fill="none" stroke="currentColor" stroke-width="3">
    <rect x="2" y="2" width="116" height="36" rx="8" />
    <path d="M 20 20 H 92 M 80 10 L 100 20 L 80 30" />
  </g>
</svg>
```

`viewBox` gives the drawing its own coordinates: four numbers for the left edge, the top edge, the width and the height of the area to show, here 120 units wide and 40 high. `width` and `height` ask for 240 by 80 CSS pixels, so each unit is drawn two pixels wide. Change those two numbers to 60 and 20 and the whole drawing shrinks with them, line widths included, because nothing inside it is measured in pixels.

The group sets the paint once. Fill, stroke and stroke width are inherited, so the frame and the arrow both take them from the `g`. The `rect` draws the frame, its corners rounded by `rx`. The path reads like instructions to a pen: `M 20 20` moves to a point without drawing, `H 92` draws a horizontal line to x = 92, the second `M` lifts the pen and puts it down at the top of the arrowhead, and the two `L` commands draw its two strokes. Upper-case commands take absolute coordinates and lower-case ones relative, so the same path can be written as `m 20 20 h 72 m -12 -10 l 20 10 l -20 10`, each step measured from where the pen stopped.

`currentColor` makes the stroke the value of the CSS `color` property, which an inline `svg` inherits from the HTML around it. Put the arrow in a link and it is drawn in the link’s colour; switch the page to a dark theme and it follows, without a second file. The `title` is not drawn. It gives the drawing its accessible name, and SVG 2 notes that browsers commonly show it as a tooltip when the pointer rests on the drawing.

The same drawing as a file of its own needs the namespace on its root element, `<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 120 40">`, and a page shows it with `<img src="/images/next.svg" alt="Next step">`. It is then a separate document. CSS inheritance runs from parent to child, and the root element of that

document has no parent, so `currentColor` inside the file no longer follows the text around the `img`. The `alt` attribute now supplies the text alternative, and SVG 2 expects the browser to show the file in secure animated mode, with no script and no external files.

This site's accessibility statement says its pages are written in HTML, CSS and SVG, with WAI-ARIA where HTML alone does not name something, and that every page is complete without executable JavaScript. There, as in this article, SVG is one of the languages a page is written in, next to its markup and its style sheets.

Where it goes wrong

A line drawing comes out as a solid shape. The initial value of `fill` is black, and the initial value of `stroke` is none. An open path meant as a line, such as an arrowhead or a tick, is filled as if its ends were joined, and with no stroke the lines themselves are not drawn at all. Set `fill="none"` and a `stroke` on line art, once, on the group that holds it.

The icon disappears in dark mode. Exported icons often carry a fixed colour, `fill="#1a1a1a"`, which stays dark on a dark background. Such an attribute is also the weakest style there is: SVG 2 gives presentation attributes a specificity of zero, so a stray CSS rule elsewhere overrides it without anyone noticing. Use `currentColor` for icons that sit in text, and set colours from CSS rather than in the markup.

The drawing will not scale, or scales wrongly. Without `viewBox` the drawing's units are the browser's pixels, so changing its width only crops or pads it, and `preserveAspectRatio` is ignored altogether. A negative width or height in `viewBox` is an error and the attribute counts as missing; a width or height of zero disables rendering. Give every drawing a `viewBox`, and let `width` and `height`, or CSS, decide the size. The default, `xMidYMid meet`, keeps the proportions and centres the drawing in its box.

Scaling is not the same as staying legible. Nothing in a vector drawing stops it being shown at 16 pixels, but detail that reads well at 200 can merge into a smudge there. A portfolio entry on building this company's visual identity describes a vector cog drawn for the browser tab on the argument that a bold silhouette survives 16 pixels. In that work, carried out in July 2026, the cog and the company mark were rendered side by side at 16, 32 and 48 pixels over light and dark tab strips: the mark stayed recognisable, the cog washed out, and the

tab icon moved to the mark. Test a drawing at the smallest size it will be shown, on both kinds of background, and draw a simpler version for small sizes when it fails.

An uploaded SVG is a document, not a picture. An SVG file can contain script and references to other files. Its media-type registration says plainly that scripting languages are executable content, and warns that maliciously constructed XML entities, expanded over and over, can use up large amounts of memory. How much of that runs depends on how the file is shown. SVG 2 expects an `object` element without sandboxing to run it in dynamic interactive mode, with script, while `img` and CSS background images use secure animated mode, with no script and no external references. Show SVG from outside sources only through `img`, serve uploads from a separate domain, or clean them against an allow-list of elements and attributes before they are stored.

Further reading

- Scalable Vector Graphics (SVG) 2 — the W3C Candidate Recommendation Snapshot of 6 October 2026, with its status, abstract and every chapter.
- Coordinate Systems, Transformations and Units — `viewBox`, `preserveAspectRatio` and the initial coordinate system.
- Paths — the path commands, absolute and relative coordinates, and what happens to path data with an error in it.
- Painting: Filling, Stroking and Marker Symbols — `fill`, `stroke`, their initial values and the `currentColor` value.
- Conformance Criteria — the five processing modes and which one each way of embedding gets.
- Document Structure — `title` and `desc`, and the implicit ARIA roles of SVG elements.

Glossary

Choose for your actual constraints

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is SVG?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Printed from https://engineer.company/glossary/svg/

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.