WIKIPEDIA

Scalable Vector Graphics

Scalable Vector Graphics (**SVG**) is an Extensible Markup Language (XML)-based vector image format for two-dimensional graphics with support for interactivity and animation. The SVG specification is an open standard developed by the World Wide Web Consortium (W3C) since 1999.

SVG images and their behaviors are defined in XML text files. This means that they can be searched, indexed, scripted, and compressed. As XML files, SVG images can be created and edited with any text editor, as well as with drawing software.

All major modern web browsers—including Mozilla Firefox, Internet Explorer, Google Chrome, Opera, Safari, and Microsoft Edge—have SVG rendering support.

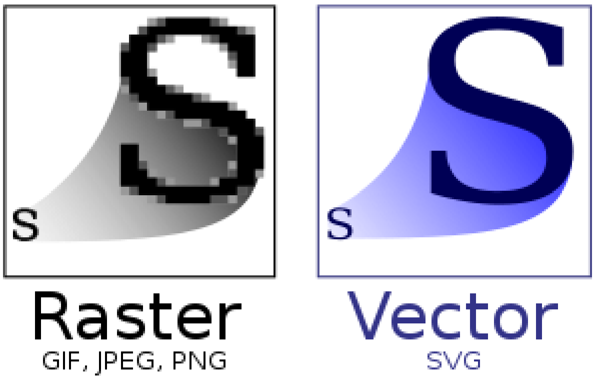
Scalable Vector Graphics

Internet media type	`image/svg+xml`[1][2]
Uniform Type Identifier (UTI)	`public.svg-image`
Developed by	W3C
Initial release	4 September 2001
Latest release	1.1 (Second Edition) (16 August 2011)
Type of format	Vector graphics
Extended from	XML
Standard	W3C SVG (https://www.w3.org/TR/SVG/)
Open format?	Yes
Website	www.w3.org/Graphics/SVG/ (https://www.w3.org/Graphics/SVG/)

Contents

Overview

SVG has been in development within the World Wide Web Consortium (W3C) since 1999 after six competing proposalsfor vector graphics languages had been submitted to the consortium during 1998. The early SVG Working Group decided not to develop any of the commercial submissions, but to create a new markup language that was informed by but not really based on any of them.[3]



This image illustrates the difference between bitmap and vector images. The bitmap image is composed of a fixed set of pixels, while the vector image is composed of a fixed set of shapes. In the picture, scaling the bitmap reveals the pixels while scaling the vector image preserves the shapes.

SVG allows three types of graphic objects: vector graphic shapes such as paths and outlines consisting of straight lines and curves, bitmap images, and text. Graphical objects can be grouped, styled, transformed and composited into previously rendered objects. The feature set includes nested transformations, clipping paths, alpha masks, filter effects and template objects. SVG drawings can be interactive and can include animation, defined in the SVG XML elements or via scripting that accesses the SVG Document Object Model (DOM). SVG uses CSS for styling and JavaScript for scripting. Text, including internationalization and localization, appearing in plain text within the SVG DOM, enhances the accessibility of SVG graphics.[4]

The SVG specification was updated to version 1.1 in 2011. There are two 'Mobile SVG Profiles,' SVG Tiny and SVG Basic, meant for mobile devices with reduced computational and display capabilities.[5] Scalable Vector Graphics 2 became a W3C Candidate Recommendation on 15 September 2016. SVG 2 incorporates several new features in addition to those of SVG 1.1 and SVG Tiny 1.2.[6]

Printing

Though the SVG Specification primarily focuses on vector graphics markup language, its design includes the basic capabilities of a page description language like Adobe's PDF. It contains provisions for rich graphics, and is compatible with CSS for styling purposes. SVG has the information needed to place each glyph and image in a chosen location on a printed page.[7]

Scripting and animation

SVG drawings can be dynamic and interactive. Time-based modifications to the elements can be described in SMIL, or can be programmed in a scripting language (e.g. JavaScript). The W3C explicitly recommends SMIL as the standard for animation in SVG.[8]

A rich set of event handlers such as "*onmouseover*" and "*onclick*" can be assigned to any SVG graphical object to apply actions and events.

Compression

SVG images, being XML, contain many repeated fragments of text, so they are well suited for lossless data compression algorithms. When an SVG image has been compressed with the gzip algorithm, it is referred to as an "SVGZ" image and uses the corresponding `.svgz` filename extension. Conforming SVG 1.1 viewers will display compressed images.[9] An SVGZ file is typically 20 to 50 percent of the original size.[10] W3C provides SVGZ files to test for conformance.[11]

Development history

SVG was developed by the W3C SVG Working Group starting in 1998, after six competing vector graphics submissions were received that year:

- Web Schematics, from CCLRC[12]
- PGML, from Adobe Systems, IBM, Netscape and Sun Microsystems[13]
- VML, by Autodesk, Hewlett-Packard, Macromedia, Microsoft, and Vision[14]
- Hyper Graphics Markup Language (HGML), by Orange UK and PRP[15]
- WebCGM, from Boeing, InterCAP Graphics Systems, Inso Corporation, CCLRC, and Xerox[16]
- DrawML, from Excosoft AB[3]

The working group was chaired at the time by Chris Lilley of the W3C.

Version 1.x

- SVG 1.0 became a W3C Recommendation on 4 September 2001.[17]
- SVG 1.1 became a W3C Recommendation on 14 January 2003.[18] The SVG 1.1 specification is modularized in order to allow subsets to be defined as profiles. Apart from this, there is very little difference between SVG 1.1 and SVG 1.0.
    - SVG Tiny and SVG Basic (the Mobile SVG Profiles) became W3C Recommendations on 14 January 2003. These are described as profiles of SVG 1.1.[19]
- SVG Tiny 1.2 became a W3C Recommendation on 22 December 2008.[20] It was initially drafted as a profile of the planned SVG Full 1.2 (which has since been dropped in favor of SVG 2),[21] but was later refactored as a standalone specification. It is generally poorly supported.
- SVG 1.1 Second Edition, which includes all the errata and clarifications, but no new features to the original SVG 1.1 was released on 16 August 2011.[4]
- SVG Tiny 1.2 Portable/Secure, A more secure subset of the SVG Tiny 1.2 profile introduced as an IETF draft standard on July 29, 2020.[22] Also known as SVG Tiny P/S. SVG Tiny 1.2 Portable/Secure is a requirement of the BIMI draft standard.[23]

Version 2.x

SVG 2.0 removes or deprecates some features of SVG 1.1 and incorporates new features from HTML5 and Web Open Font Format:[24]

- For example, SVG 2.0 removes several font elements such as `glyph` and `altGlyph` (replaced by the WOFF font format).
- The `xml:space` attribute is deprecated in favor of CSS.
- HTML5 features such as `translate` and `data-*` attributes have been added.
- Text handling features from SVG Tiny 1.2 are annotated as to be included, but not yet formalized in text.[25] Some other 1.2 features are cherry-picked in,[24] but SVG 2.0 is not a superset of SVG tiny 1.2 in general.

It reached Candidate Recommendation stage on 15 September 2016. The latest draft was released on 26 May 2020.[26]

Mobile profiles

Because of industry demand, two mobile profiles were introduced with SVG 1.1: *SVG Tiny* (SVGT) and *SVG Basic* (SVGB).

These are subsets of the full SVG standard, mainly intended for user agents with limited capabilities. In particular, SVG Tiny was defined for highly restricted mobile devices such as cellphones; it does not support styling or scripting.[27] SVG Basic was defined for higher-level mobile devices, such as smartphones.

In 2003, the 3GPP, an international telecommunications standards group, adopted SVG Tiny as the mandatory vector graphics media format for next-generation phones. SVGT is the required vector graphics format and support of SVGB is optional for Multimedia Messaging Service (MMS) and Packet-switched Streaming Service.[28][29][30] It was later added as required format for vector graphics in 3GPP IP Multimedia Subsystem (IMS).[31][32]

Differences from non-mobile SVG

Neither mobile profile includes support for the full Document Object Model (DOM), while only SVG Basic has optional support for scripting, but because they are fully compatible subsets of the full standard, most SVG graphics can still be rendered by devices which only support the mobile profiles.[33]

SVGT 1.2 adds a microDOM (μDOM), styling and scripting.[27]

Related work

The MPEG-4 Part 20 standard - *Lightweight Application Scene Representation (LASeR) and Simple Aggregation Format (SAF)* is based on SVG Tiny.[34] It was developed by MPEG (ISO/IEC JTC1/SC29/WG11) and published as ISO/IEC 14496-20:2006.[35] SVG capabilities are enhanced in MPEG-4 Part 20 with key features for mobile services, such as dynamic updates, binary encoding, state-of-art font representation.[36] SVG was also accommodated in MPEG-4 Part 11, in the Extensible MPEG-4 Textual (XMT) format - a textual representation of the MPEG-4 multimedia content using XML.[37]

Functionality

The SVG 1.1 specification defines 14 functional areas or feature sets:[18]

Paths

Simple or compound shape outlines are drawn with curved or straight lines that can be filled in, outlined, or used as a clipping path. Paths have a compact coding.
For example, `M` (for "move to") precedes initial numeric *x* and *y* coordinates, and `L` (for "line to") precedes a point to which a line should be drawn. Further command letters (`C`, `S`, `Q`, `T`, and `A`) precede data that is used to draw various Bézier and elliptical curves. `Z` is used to close a path.
In all cases, absolute coordinates follow capital letter commands and relative coordinates are used after the equivalent lower-case letters.[38]

Basic shapes

Straight-line paths and paths made up of a series of connected straight-line segments (polylines), as well as closed polygons, circles, and ellipses can be drawn. Rectangles and round-cornered rectangles are also standard elements.[39]

Text

Unicode character text included in an SVG file is expressed as XML character data. Many visual effects are possible, and the SVG specification automatically handles bidirectional text (for composing a combination of English and Arabic text, for example), vertical text (as Chinese was historically written) and characters along a curved path (such as the text around the edge of the Great Seal of the United States).[40]

Painting

SVG shapes can be filled and outlined (painted with a color, a gradient, or a pattern). Fills may be opaque, or have any degree of transparency.

"Markers" are line-end features, such as arrowheads, or symbols that can appear at the vertices of a polygon.[41]

Color

Colors can be applied to all visible SVG elements, either directly or via `fill`, `stroke`, and other properties. Colors are specified in the same way as in CSS2, i.e. using names like `black` or `blue`, in hexadecimal such as `#2f0` or `#22ff00`, in decimal like `rgb(255,255,127)`, or as percentages of the form `rgb(100%,100%,50%)`.[42]

Gradients and patterns

SVG shapes can be filled or outlined with solid colors as above, or with color gradients or with repeating patterns. Color gradients can be linear or radial (circular), and can involve any number of colors as well as repeats. Opacity gradients can also be specified. Patterns are based on predefined raster or vector graphic objects, which can be repeated in `x` and `y` directions. Gradients and patterns can be animated and scripted.[43]

Since 2008, there has been discussion[44][45] among professional users of SVG that either gradient meshes or preferably diffusion curves could usefully be added to the SVG specification. It is said that a "simple representation [using diffusion curves] is capable of representing even very subtle shading effects"[46] and that "Diffusion curve images are comparable both in quality and coding efficiency with gradient meshes, but are simpler to create (according to several artists who have used both tools), and can be captured from bitmaps fully automatically."[47] The current draft of SVG 2 includes gradient meshes.[48]

Clipping, masking and compositing

Graphic elements, including text, paths, basic shapes and combinations of these, can be used as outlines to define both *inside* and *outside* regions that can be painted (with colors, gradients and patterns) independently. Fully opaque *clipping paths* and semi-transparent *masks* are *composited* together to calculate the color and opacity of every pixel of the final image, using alpha blending.[49]

Filter effects[50]

A filter effect consists of a series of graphics operations that are applied to a given source vector graphic to produce a modified bitmapped result.

Interactivity

SVG images can interact with users in many ways. In addition to hyperlinks as mentioned below, any part of an SVG image can be made receptive to user interface events such as changes in focus, mouse clicks, scrolling or zooming the image and other pointer, keyboard and document events. Event handlers may start, stop or alter animations as well as trigger scripts in response to such events.[51]

Linking

SVG images can contain hyperlinks to other documents, using XLink. Through the use of the `<view>` element or a fragment identifier, URLs can link to SVG files that change the visible area of the document. This allows for creating specific view states that are used to zoom in/out of a specific area or to limit the view to a specific element. This is helpful when creating sprites. XLink support in combination with the `<use>` element also allow linking to and re-using internal and external elements. This allows coders to do more with less markup and makes for cleaner code.[52]

Scripting

All aspects of an SVG document can be accessed and manipulated using scripts in a similar way to HTML. The default scripting language is JavaScript and there are defined Document

Object Model (DOM) objects for every SVG element and attribute. Scripts are enclosed in `<script>` elements. They can run in response to pointer events, keyboard events and document events as required.[53]

Animation

SVG content can be animated using the built-in animation elements such as `<animate>`, `<animateMotion>` and `<animateColor>`. Content can be animated by manipulating the DOM using ECMAScript and the scripting language's built-in timers. SVG animation has been designed to be compatible with current and future versions of Synchronized Multimedia Integration Language (SMIL). Animations can be continuous, they can loop and repeat, and they can respond to user events, as mentioned above.[54]

Fonts

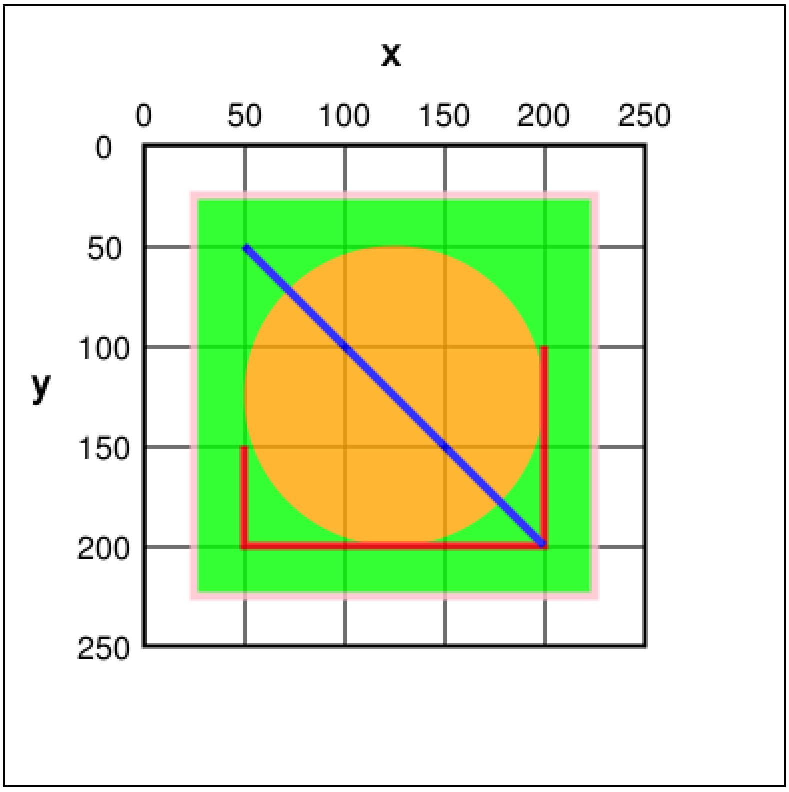
As with HTML and CSS, text in SVG may reference external font files, such as system fonts. If the required font files do not exist on the machine where the SVG file is rendered, the text may not appear as intended. To overcome this limitation, text can be displayed in an *SVG font*, where the required glyphs are defined in SVG as a font that is then referenced from the `<text>` element.[55]

Metadata

In accord with the W3C's Semantic Web initiative, SVG allows authors to provide metadata about SVG content. The main facility is the `<metadata>` element, where the document can be described using Dublin Core metadata properties (e.g. title, creator/author, subject, description, etc.). Other metadata schemas may also be used. In addition, SVG defines `<title>` and `<desc>` elements where authors may also provide plain-text descriptive material within an SVG image to help indexing, searching and retrieval by a number of means.[56]

An SVG document can define components including shapes, gradients etc., and use them repeatedly. SVG images can also contain raster graphics, such as PNG and JPEG images, and further SVG images.

Example

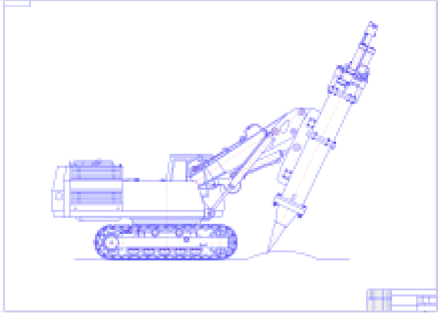


This code will produce the colored shapes shown in the image, excluding the grid and labels:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg width="391" height="391" viewBox="-70.5 -70.5 391 391" xmlns="http://www.w3.org/2000/svg">
<rect fill="#fff" stroke="#000" x="-70" y="-70" width="390" height="390"/>
<g opacity="0.8">
<rect x="25" y="25" width="200" height="200" fill="lime" stroke-width="4" stroke="pink" />
    <circle cx="125" cy="125" r="75" fill="orange" />
    <polyline points="50,150 50,200 200,200 200,100" stroke="red" stroke-width="4" fill="none"
/>
    <line x1="50" y1="50" x2="200" y2="200" stroke="blue" stroke-width="4" />
</g>
</svg>
```

SVG on the web

The use of SVG on the web was limited by the lack of support in older versions of Internet Explorer (IE). Many web sites that serve SVG images, such as Wikipedia, also provide the images in a raster format, either automatically by HTTP content negotiation or by allowing the user directly to choose the file.



SVG exported from KOMPAS-Graphic

Google announced on 31 August 2010 that it had started to index SVG content on the web, whether it is in standalone files or embedded in HTML, and that users would begin to see such content listed among their search results.[57] It was announced on 8 December 2010 that Google Image Search would also begin indexing SVG files.[58] The site announced an option to restrict image searches to SVG files on 11 February 2011.[59]

Native browser support

Konqueror was the first browser to support SVG in release version 3.2 in February 2004.[60] As of 2011, all major desktop browsers, and many minor ones, have some level of SVG support. Other browsers' implementations are not yet complete; see comparison of layout engines for further details.

Some earlier versions of Firefox (e.g. versions between 1.5 and 3.6[61]), as well as a smattering of other now-outdated web browsers capable of displaying SVG graphics, needed them embedded in `<object>` or `<iframe>` elements to display them integrated as parts of an HTML webpage instead of using the standard way of integrating images with `<img>`.[62] However, SVG images may be included in XHTML pages using XML namespaces.[63]

Tim Berners-Lee, the inventor of the World Wide Web, was critical of early versions of Internet Explorer for its failure to support SVG.[64]

- Opera (since 8.0) has support for the SVG 1.1 Tiny specification, while Opera 9 includes SVG 1.1 Basic support and some of SVG 1.1 Full. Opera 9.5 has partial SVG Tiny 1.2 support. It also supports SVGZ (compressed SVG).
- Browsers based on the Gecko layout engine (such as Firefox, Flock, Camino, and SeaMonkey) all have had incomplete support for the SVG 1.1 Full specification since 2005. The Mozilla site has an overview of the modules which are supported in Firefox[65] and of the modules which are in development.[66] Gecko 1.9, included in Firefox 3.0, adds support for more of the SVG specification (including filters).[67]
- Pale Moon, which uses the Goanna layout engine (a fork of the Gecko engine), supports SVG.

- Browsers based on WebKit (such as Apple's Safari, Google Chrome, and The Omni Group's OmniWeb) have had incomplete support for the SVG 1.1 Full specification since 2006.[68]
- Amaya has partial SVG support.
- Internet Explorer 8 and older versions do not support SVG.[69][70] IE9 (released 14 March 2011) supports the basic SVG feature set.[71] IE10 extended SVG support by adding SVG 1.1 filters.[72]
- Microsoft Edge supports SVG 1.1.[73]
- The Maxthon Cloud Browser also supports SVG.

There are several advantages to native and full support: plugins are not needed, SVG can be freely mixed with other content in a single document, and rendering and scripting become considerably more reliable.[74]

Mobile support

SVG Tiny (SVGT) 1.1 and 1.2 are mobile profiles for SVG. SVGT 1.2 includes some features not found in SVG 1.1, including non-scaling strokes, which are supported by some SVG 1.1 implementations, such as Opera, Firefox and WebKit. As shared code bases between desktop and mobile browsers increased, the use of SVG 1.1 over SVGT 1.2 also increased.

Support for SVG may be limited to SVGT on older or more limited smart phones or may be primarily limited by their respective operating system. Adobe Flash Lite has optionally supported SVG Tiny since version 1.1. At the SVG Open 2005 conference, Sun demonstrated a mobile implementation of SVG Tiny 1.1 for the Connected Limited Device Configuration (CLDC) platform.[75]

Mobiles that use Opera Mobile, as well as the iPhone's built in browser, also include SVG support. However, even though it used the WebKit engine, the Android built-in browser did not support SVG prior to v3.0 (Honeycomb).[76] Prior to v3.0, Firefox Mobile 4.0b2 (beta) for Android was the first browser running under Android to support SVG by default.[77]

The level of SVG Tiny support available varies from mobile to mobile, depending on the SVG engine installed. Many newer mobile products support additional features beyond SVG Tiny 1.1, like gradient and opacity; this is sometimes referred to as "SVGT 1.1+", though there is no such standard.

RIM's BlackBerry has built-in support for SVG Tiny 1.1 since version 5.0.[78] Support continues for WebKit-based BlackBerry Torch browser in OS 6 and 7.[79]

Nokia's S60 platform has built-in support for SVG. For example, icons are generally rendered using the platform's SVG engine. Nokia has also led the JSR 226: Scalable 2D Vector Graphics API expert group that defines Java ME API for SVG presentation and manipulation. This API has been implemented in S60 Platform 3rd Edition Feature Pack 1 and onward.[80] Some Series 40 phones also support SVG (such as Nokia 6280).

Most Sony Ericsson phones beginning with K700 (by release date) support SVG Tiny 1.1. Phones beginning with K750 also support such features as opacity and gradients. Phones with Sony Ericsson Java Platform-8 have support for JSR 226.

Windows Phone has supported SVG since version 7.5.

SVG is also supported on various mobile devices from Motorola, Samsung, LG, and Siemens mobile/BenQ-Siemens. eSVG, an SVG rendering library mainly written for embedded devices, is available on some mobile platforms.[81][82]

Application support

SVG images can be produced by the use of a vector graphics editor, such as Inkscape, Adobe Illustrator, Adobe Flash Professional, or CorelDRAW, and rendered to common raster image formats such as PNG using the same software. Additionally, editors like Inkscape and Boxy SVG provide tools to trace raster images to Bézier curves typically using image tracing back-ends like potrace,[83][84] autotrace (https://github.com/autotrace/autotrace), and imagetracerjs (https://github.com/jankovicsandras/imagetracerjs).

Software can be programmed to render SVG images by using a library such as librsvg used by GNOME since 2000, or Batik. SVG images can also be rendered to any desired popular image format by using ImageMagick, a free command-line utility (which also uses librsvg under the hood).

Other uses for SVG include embedding for use in word processing (e.g. with LibreOffice) and desktop publishing (e.g. Scribus), plotting graphs (e.g. gnuplot), and importing paths (e.g. for use in GIMP or Blender). Microsoft 365 and Microsoft Office 2019 offer support for exporting,[85] importing and editing SVG images. The Uniform Type Identifier for SVG used by Apple is `public.svg-image` and conforms to `public.image` and `public.xml`.

See also

- Canvas element
- Comparison of graphics file formats
- Comparison of raster-to-vector conversion software
- Comparison of vector graphics editors
- Computer graphics
- Computer Graphics Metafile
- Image file formats
- Resolution independence
- SVG files in MediaWiki on Commons: Help:SVG

Scalable Vector Graphics

- sXBL
- SVG Working Group
- SVG filter effects
- Precision Graphics Markup Language
- Vector Markup Language
- XHTML+MathML+SVG
- SVG animation
- Comparison of layout engines (SVG)

References

1. "Media Type Registration for image/svg+xml" (http://www.w3.org/TR/SVG/mimereg.html). W3C. Retrieved 5 February 2014.
2. "XML Media Types" (http://tools.ietf.org/html/rfc3023#section-8.19). Retrieved 5 February 2014.
3. "Secret Origin of SVG" (http://www.w3.org/Graphics/SVG/WG/wiki/Secret_Origin_of_SVG). World Wide Web Consortium. 21 December 2007. Retrieved 1 January 2011.
4. "Scalable Vector Graphics (SVG) 1.1 (Second Edition)" (http://www.w3.org/TR/2011/REC-SVG11-20110816/). W3C.
5. Capin, Tolga. "Mobile SVG Profiles: SVG Tiny and SVG Basic" (https://www.w3.org/TR/SVGMobile/). W3C. Retrieved 28 January 2017.
6. "Scalable Vector Graphics (SVG) 2" (https://www.w3.org/TR/SVG2/). W3C. Retrieved 28 January 2017.

7. Alex, Danilo; Fujisawa, Jun (2002). "SVG as a Page Description Language" (http://www.svgopen.org/2002/papers/danilo_fujisawa__svg_as_page_description_language/index.html). svgopen.org. Retrieved 24 February 2010.
8. Paul, Festa (9 January 2003). "W3C releases scripting standard, caveat" (http://news.cnet.com/2100-1023-979976.html). CNet. Retrieved 24 February 2010.
9. See www.w3.org/TR/SVG11/conform.html#ConformingSVGViewers (http://www.w3.org/TR/SVG11/conform.html#ConformingSVGViewers) which states, "SVG implementations must correctly support gzip-encoded [RFC1952] and deflate-encoded [RFC1951] data streams, for any content type (including SVG, script files, images)."
10. "Saving compressed SVG (SVGZ)" (https://www.adobe.com/svg/illustrator/compressedsvg.html). *SVG Zone*. Adobe Systems. 14 July 2009. Retrieved 24 February 2010.
11. For example, dev.w3.org/SVG/profiles/1.1F2/test/harness/htmlObject/conform-viewers-01-t.html (http://dev.w3.org/SVG/profiles/1.1F2/test/harness/htmlObject/conform-viewers-01-t.html)
12. "Schematic Graphics" (https://www.w3.org/TR/1998/NOTE-WebSchematics-19980331/Overview.html). *www.w3.org*.
13. Al-Shamma, Nabeel; Robert Ayers; Richard Cohn; Jon Ferraiolo; Martin Newell; Roger K. de Bry; Kevin McCluskey; Jerry Evans (10 April 1998). "Precision Graphics Markup Language (PGML)" (http://www.w3.org/TR/1998/NOTE-PGML-19980410). W3C. Retrieved 8 May 2009.
14. Mathews, Brian; Brian Dister; John Bowler; Howard Cooper stein; Ajay Jindal; Tuan Nguyen; Peter Wu; Troy Sandal (13 May 1998). "Vector Markup Language (VML)" (http://www.w3.org/TR/1998/NOTE-VML-19980513). W3C. Retrieved 8 May 2009.
15. "Hyper Graphics Markup Language (HGML)" (https://www.w3.org/TR/NOTE-HGML.html). *www.w3.org*.
16. "WebCGM Profile" (http://xml.coverpages.org/NOTE-WebCGM-19981104.html). *xml.coverpages.org*.
17. Ferraiolo, Jon (4 September 2001). "Scalable Vector Graphics (SVG) 1.0 Specification" (http://www.w3.org/TR/SVG10/). World Wide Web Consortium. Retrieved 24 February 2010.
18. Ferraiolo, Jon (16 August 2011). "Scalable Vector Graphics (SVG) 1.1 Specification" (http://www.w3.org/TR/SVG11/). World Wide Web Consortium. Retrieved 14 April 2016.
19. Capin, Tolga (15 June 2009). "Mobile SVG Profiles: SVG Tiny and SVG Basic" (http://www.w3.org/TR/2003/REC-SVGMobile-20030114/). World Wide Web Consortium. Retrieved 24 October 2010.

20. Andersson, Ola (22 December 2008). "Scalable Vector Graphics (SVG) Tiny 1.2 Specification" (http://www.w3.org/TR/SVGTiny12/). World Wide Web Consortium. Retrieved 24 February 2010.
21. Dengler, Patrick (8 July 2010). "Getting to SVG 2.0: A report from the SVG Working Group Face-to-Face (May 24th – June 1st 2010)" (https://blogs.msdn.microsoft.com/ie/2010/06/08/getting-to-svg-2-0-a-report-from-the-svg-working-group-face-to-face-may-24th-june-1st-2010/). *Microsoft Developer Network*. Microsoft. Retrieved 26 August 2010.
22. "SVG Tiny Portable/Secure" (https://tools.ietf.org/html/draft-svg-tiny-ps-abrotman-00.html). *IETF - Internet Engineering Task Force*. Retrieved 9 February 2021.
23. "Implementation Guide" (https://bimigroup.org/implementation-guide/). *BIMI Group*. Retrieved 9 February 2021.
24. "Appendix K: Changes from SVG 1.1" (https://svgwg.org/svg2-draft/changes.html). World Wide Web Consortium. 26 May 2020. Retrieved 31 December 2020.
25. "Chapter 11: Text" (https://svgwg.org/svg2-draft/text.html). World Wide Web Consortium. 26 May 2020. Retrieved 31 December 2020.
26. "Scalable Vector Graphics (SVG) 2" (https://svgwg.org/svg2-draft/). World Wide Web Consortium. 26 May 2020. Retrieved 13 September 2020.
27. "SVG Tiny 1.2" (http://www.w3.org/TR/SVGTiny12/intro.html#SVGTiny12). *Scalable Vector Graphics (SVG) Tiny 1.2 Specification*. W3C. 22 December 2008. Retrieved 5 March 2014.
28. "SVG in 3GPP Multimedia Messaging and Streaming Services (version March 2003)" (http://www.svgopen.org/2003/paperAbstracts/svgmms.htm). SVG Open. 2003. Retrieved 19 October 2009.
29. "3GPP Multimedia Messaging Service (MMS); Media formats and codecs (Release 5); 3GPP TS 26.140 V5.2.0 (2002-12); Technical Specification" (https://www.3gpp.org/ftp/Specs/archive/26_series/26.140/26140-520.zip) (zipped doc). 3GPP. 2 January 2003. Retrieved 25 February 2010.
30. "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Multimedia Messaging Service (MMS); Media formats and codecs (Release 5)" (https://www.3gpp.org/ftp/Specs/archive/26_series/26.234/26234-540.zip) (zipped doc). *3GPP TS 26.140 V5.2.0 (2002-12)*. 3GPP. March 2003. Retrieved 24 February 2010.
31. "3GPP Specification detail - 3GPP TS 26.141: IP Multimedia System (IMS) Messaging and Presence; Media formats and codecs" (http://www.3gpp.org/ftp/Specs/html-info/26141.htm). 3GPP. 10 December 2009. Retrieved 24 February 2010.

32. "Building Interactive Websites using Scalable Vector Graphics (SVG) – [With Examples]" (https://www.intelegain.com/scalable-vector-graphics/). intelegain.com. Archived from the original (https://www.intelegain.com) on 21 November 2018. Retrieved 21 November 2018.
33. "Listing of phones that support SVG" (http://webarchive.loc.gov/all/20100108141222/http://www.svg.org/special/svg_phones). Svg.org. Archived from the original (http://svg.org/special/svg_phones) on 8 January 2010. Retrieved 24 October 2010.
34. Concolato, Cyril (July 2005). "MPEG-4 LASeR white paper" (http://mpeg.chiariglione.org/technologies/mpeg-4/mp04-lsr/index.htm). International Organization for Standardization. Retrieved 29 August 2010.
35. "ISO/IEC 14496-20:2008 - Information technology – Coding of audio-visual objects – Part 20: Lightweight Application Scene Representation (LASeR) and Simple Aggregation Format (SAF)" (http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=52454). International Organization for Standardization. Retrieved 30 October 2009.
36. "LASeR - the MPEG standard for Rich Media Services" (http://www.mpeg-laser.org/documents/LASeRWhitePaper.pdf) (PDF). Retrieved 11 April 2010.
37. Chiariglione, Leonardo (8 March 2005). "Riding the media bits - Bits and bytes" (https://web.archive.org/web/20100925220237/http://ride.chiariglione.org/bits_and_bytes/bits_and_bytes.htm). Archived from the original (http://ride.chiariglione.org/bits_and_bytes/bits_and_bytes.htm) on 25 September 2010. Retrieved 30 October 2009.
38. "SVG specification, "Paths" " (http://www.w3.org/TR/SVG11/paths.html). World Wide Web Consortium. 14 January 2003. Retrieved 19 October 2009.
39. "SVG specification, "Basic Shapes" " (http://www.w3.org/TR/SVG11/shapes.html). World Wide Web Consortium. 14 January 2003. Retrieved 19 October 2009.
40. "SVG specification, "Text" " (http://www.w3.org/TR/SVG11/text.html). World Wide Web Consortium. 14 January 2003. Retrieved 19 October 2009.
41. "SVG specification, "Painting: Filling, Stroking and Marker Symbols" " (http://www.w3.org/TR/SVG11/painting.html). World Wide Web Consortium. 14 January 2003. Retrieved 19 October 2009.
42. "SVG specification, "Color" " (http://www.w3.org/TR/SVG11/color.html). World Wide Web Consortium. 14 January 2003. Retrieved 19 October 2009.
43. "SVG specification, "Gradients and Patterns" " (http://www.w3.org/TR/SVG11/pservers.html). World Wide Web Consortium. 14 January 2003. Retrieved 19 October 2009.

44. "IRC log of svg on 2008-11-17" (http://www.w3.org/2008/11/17-svg-irc). W3C. 2008. Retrieved 25 May 2009. "DS: Priorities should be layout, diffusion curves, 2.5D and make sure we work well with CSS on that"
45. "Propositions for the next SVG spec" (http://lists.w3.org/Archives/Public/www-svg/2009Mar/0084.html). W3C. 16 March 2009. Retrieved 25 May 2009. "Indeed, we do plan to add new gradient capabilities to SVG in the next version, and we are looking into diffusion curves, which I think will meet your needs ... *Diffusion curves* seems really a great idea."
46. Rusin, Zack (2008). "SVG in KDE: Freedom of Beauty" (http://www.svgopen.org/2008/papers/104-SVG_in_KDE/). SVG Open. Retrieved 25 May 2009.
47. Orzan, Alexandrina; Bousseau, Adrien; Winnemöller, Holger; Barla, Pascal; Thollot, Joëlle; Salesin, David (2008). "Diffusion Curves: A Vector Representation for Smooth-Shaded Images" (http://artis.imag.fr/Publications/2008/OBWBTS08/diffusion_curves.pdf) (PDF). *ACM Transactions on Graphics*. **27**. doi:10.1145/1360612.1360691 (https://doi.org/10.1145%2F1360612.1360691). Retrieved 25 May 2009.
48. "Scalable Vector Graphics (SVG) 2, W3C Editor's Draft 13 June 2013" (https://svgwg.org/svg2-draft/pservers.html#MeshGradients). SVG WG. 2013. Retrieved 17 September 2013.
49. "SVG specification, "Clipping, Masking and Compositing" " (http://www.w3.org/TR/SVG11/masking.html). World Wide Web Consortium. 14 January 2003. Retrieved 19 October 2009.
50. "SVG specification, "Filter Effects" " (http://www.w3.org/TR/SVG11/filters.html). World Wide Web Consortium. 14 January 2003. Retrieved 19 October 2009.
51. "SVG specification, "Interactivity" " (http://www.w3.org/TR/SVG11/interact.html). World Wide Web Consortium. 14 January 2003. Retrieved 19 October 2009.
52. "SVG specification, "Linking" " (http://www.w3.org/TR/SVG11/linking.html). World Wide Web Consortium. 14 January 2003. Retrieved 19 October 2009.
53. "SVG specification, "Scripting" " (http://www.w3.org/TR/SVG11/script.html). World Wide Web Consortium. 14 January 2003. Retrieved 19 October 2009.
54. "SVG specification, "Animation" " (http://www.w3.org/TR/SVG11/animate.html). World Wide Web Consortium. 14 January 2003. Retrieved 19 October 2009.
55. "SVG specification, "Fonts" " (http://www.w3.org/TR/SVG11/fonts.html). World Wide Web Consortium. 14 January 2003. Retrieved 19 October 2009.
56. "SVG specification, "Metadata" " (http://www.w3.org/TR/SVG11/metadata.html). World Wide Web Consortium. Retrieved 19 October 2009.

57. Stanescu, Bogdan; Sarapata, John (31 August 2010). "Google now indexes SVG" (http://googlewebmastercentral.blogspot.com/2010/08/google-now-indexes-svg.html). *Webmaster central blog*. Google, Inc. Retrieved 1 September 2010.
58. "googleimages" (https://www.twitter.com/googleimages/status/12589924704329728). Google, Inc. 8 December 2010. Retrieved 17 February 2011.
59. Singhal, Amit (11 February 2011). "This week in search 2/11/11" (http://googleblog.blogspot.com/2011/02/this-week-in-search-21111.html). Google, Inc. Retrieved 17 February 2011.
60. Streichardt, Andreas (16 September 2003). "KDE Conquers the Vectors with KSVG" (http://dot.kde.org/2003/09/16/kde-conquers-vectors-ksvg). *KDE News*. Retrieved 11 February 2012.
61. "Bug 276431 - external SVG not loaded from img tag" (https://bugzilla.mozilla.org/show_bug.cgi?id=276431). Mozilla{{inconsistent citations}}
62. Lusotec (25 February 2009). "SVG image not shown by Firefox" (https://groups.google.com/group/mozilla.dev.tech.svg/browse_thread/thread/75f14ac7244b0fa5/a6ccf7261c5110b7?lnk=raot). *mozilla.dev.tech.svg*. Google Groups. Retrieved 25 February 2010.
63. Brettz9 (25 August 2008). "SVG In HTML Introduction" (https://developer.mozilla.org/en/SVG_In_HTML_Introduction). *Mozilla Developer Center*. Mozilla. Retrieved 25 February 2010.
64. Svensson, Peter (10 September 2008). "Creator of Web spots a flaw in Internet Explorer" (http://www.nbcnews.com/id/26646919). NBC News. Associated Press. Retrieved 25 February 2010.
65. "SVG in Firefox" (https://developer.mozilla.org/en/docs/SVG_in_Firefox). *Mozilla Developer Center*. Mozilla. 23 October 2009. Retrieved 25 February 2010.
66. "Mozilla SVG Status" (https://www.mozilla.org/projects/svg/status.html). Mozilla. Retrieved 25 February 2010.
67. Mgjbot (15 January 2008). "SVG improvements in Firefox 3" (https://developer.mozilla.org/en/docs/SVG_improvements_in_Firefox_3). *Mozilla Developer Center*. Mozilla. Retrieved 25 February 2010.
68. "WebKit SVG Status" (http://webkit.org/projects/svg/status.xml). WebKit. 5 January 2010. Retrieved 25 February 2010.
69. "Windows Internet Explorer 8 Expert Zone Chat (19 June 2008)" (http://www.microsoft.com/windowsxp/expertzone/chats/transcripts/08_0619_ez_ie8.mspx). Microsoft. 19 June 2008. Retrieved 24 October 2010.
70. Schiller, Jeff (13 February 2010). "SVG Support" (http://www.codedread.com/svg-support.php). Retrieved 25 February 2010.

71. "What's New in Internet Explorer 9" (http://msdn.microsoft.com/en-us/library/ff974378(VS.85).aspx). *Microsoft Developer Network*. Microsoft. 22 March 2011. Retrieved 22 March 2011.
72. "SVG (Windows)" (https://msdn.microsoft.com/en-us/library/ie/hh673562(v=vs.85).aspx). *Internet Explorer Dev Center*. Microsoft. Retrieved 27 August 2014.
73. "Microsoft Edge web platform features status and roadmap - Microsoft Edge Development" (https://developer.microsoft.com/en-us/microsoft-edge/platform/status/). *developer.microsoft.com*. Retrieved 19 April 2018.
74. "Get ready for plug-in free browsing" (https://msdn.microsoft.com/en-us/library/ie/hh968248.aspx). MSDN. 2014. Retrieved 5 March 2014.
75. "SVG Open 2005 Conference and Exhibition - Proceedings - Cartoon Oriented User Interfaces" (http://www.svgopen.org/2005/proceedings.html#paper108). svgopen.org. Retrieved 29 August 2010.
76. "Browser doesn't render SVG images" (https://code.google.com/p/android/issues/detail?id=1376). Retrieved 1 January 2011.
77. "Firefox Mobile 4.0b2 (beta) for Android supports SVG images" (https://code.google.com/p/android/issues/detail?id=1376#c30). *Android Issue 1376*. 16 December 2010. Retrieved 29 January 2014.
78. "Support for SVG" (http://docs.blackberry.com/en/developers/deliverables/11844/Native_SVG_support_887737_11.jsp). RIM. Retrieved 6 October 2011.
79. "BlackBerry Torch The HTML5 Developer Scorecard" (http://www.sencha.com/blog/blackberry-torch-the-html5-developer-scorecard). Sencha. Retrieved 6 October 2011.
80. "S60 Product info" (https://web.archive.org/web/20071013152026/http://www.s60.com/business/productinfo/applicationsandtechnologies/java/supportedfeaturesin3rdedition). S60.com. Archived from the original (http://www.s60.com/business/productinfo/applicationsandtechnologies/java/supportedfeaturesin3rdedition) on 13 October 2007. Retrieved 19 October 2009.
81. Jezic, Damir (15–17 July 2002). "SVG for Embedded Systems" (http://www.svgopen.org/2002/papers/jezic__esvg/). *Zurich 2002*. svgopen.org. Retrieved 5 April 2012.
82. "eSVG" (http://www.intesis.hr/products/esvg). *Intesis Intelligent Embedded Software*. 2010. Retrieved 29 January 2014.
83. Selinger, Peter (2001–2013). "Potrace – Transforming bitmaps into vector graphics" (http://potrace.sourceforge.net/). *SourceForge project Potrace*. Retrieved 29 January 2014.
84. "Vectorize/trace" (http://wiki.inkscape.org/wiki/index.php/Tools#Vectorize.2Ftrace). *Inkscape wiki*. 2012. Retrieved 29 January 2014.

85. "What's new in PowerPoint for Microsoft 365" (https://support.microsoft.com/en-us/office/what-s-new-in-powerpoint-for-microsoft-365-e8ef980c-5b12-4fff-ae3f-0819e6a21a1f). *support.microsoft.com*. September 2019. Archived (https://web.archive.org/web/20210313183903/https://support.microsoft.com/en-us/office/what-s-new-in-powerpoint-for-microsoft-365-e8ef980c-5b12-4fff-ae3f-0819e6a21a1f) from the original on 13 March 2021. Retrieved 17 April 2021.

External links

- W3C SVG page (http://www.w3.org/Graphics/SVG) specifications, list of implementations
- W3C SVG primer (http://www.w3.org/Graphics/SVG/IG/resources/svgprimer.html) W3C Primer (draft) under auspices of SVG Interest Group
- Scalable Vector Graphics (https://curlie.org/Computers/Data_Formats/Graphics/Vector/SVG) at Curlie

Retrieved from "https://en.wikipedia.org/w/index.php?title=Scalable_Vector_Graphics&oldid=1018980456"

This page was last edited on 20 April 2021, at 22:03 (UTC).