

Using SVG with CSS3 and HTML5 — Supplementary Material

Example code and online extras for the [O'Reilly Media book](#) by Amelia Bellamy-Royds, Kurt Cagle, and Dudley Storey.

[Book Home](#)

[All SVG Reference Guides](#)

Units for Measurements

On this page:

[Introduction](#)

[Absolute Length Units](#)

[Font-Relative Length Units](#)

[Viewport Length Units](#)

[Percentage Lengths](#)

[Angle Units](#)

[Time Units](#)

The length units you can use to specify positions and coordinates in SVG can be separated into absolute units, which have explicit conversion factors, and relative units, which depend on other properties of the layout. As discussed in the text, the newer CSS units should be used with caution for SVG.

Percentage units have special meaning depending on the attribute or style property. However, most SVG length values use a common interpretation of percentages, that depends on whether the length is horizontal, vertical, or other.

Units are also defined for angle measurements and time measurements.

Absolute Length Units

The following absolute length units are defined in CSS 2 and SVG 1 and 1.1. They should be supported in nearly all SVG implementations. However, the relationship between the px unit and the other absolute units was defined in CSS 2.1; in older software, the relationship between px and other units will fluctuate according to the screen resolution.

px

Pixel units, directly equivalent to SVG user units.

- For print, a px should be equal to $\frac{1}{96}$ th of an inch.
- For screens, a px should represent approximately the same distance in the user's field of view (the same visual angle) as $\frac{1}{96}$ th of an inch at arm's length.
- The px unit can be adjusted to represent an even number of actual screen pixels.

in

Inches.

- 1in = 96px or user units
- 1in = 2.54cm
- In modern software, inches and all other absolute units will be adjusted to match the px unit.

cm

Centimeters.

- 1cm \cong 37.795px or user units

mm

Millimeters.

- 1mm \cong 3.7795px or user units

pt

Points.

- 1pt \cong 1.3333px or user units (1px = 0.75pt)
- 1pt = $\frac{1}{72}$ in

pc

Picas.

- 1pc = 16px or user units
- 1pc = $\frac{1}{6}$ in

Font-Relative Length Units

The following units describe lengths proportional to the size and properties of the font set on the element. They are defined in CSS 2 and SVG 1 and 1.1, and should be well supported for SVG. However, some SVG viewers do not apply a reasonable default font size for SVG, so test carefully—font-relative units may be unusually small or large if you don't specify `font-size` yourself!

em

Em units. Equivalent to the computed font-size in effect for an element.

ex

Ex units. Equivalent to the height of a lower-case letter in the font (and font-size) in effect for an element. If the font doesn't include lower-case letters, or doesn't include the metadata about the ex-height, then $1\text{ex} = 0.5\text{em}$.

The following additional font-relative units are defined in the CSS 3 Values and Units specification; they may not be supported correctly in some SVG software.

ch

Character units. Equivalent to the width of the 0 (zero) character in the font and font-size in effect for an element.

rem

Root-em units. The font size of the root element in the document, regardless of any settings for the current element.

When styles defined with relative lengths are inherited, it is the computed value of the length that is inherited, not the relationship to the current font. In other words, the value is not affected if the child element has a different font or font-size.

Viewport Length Units

The CSS 3 Values and Units specification also defines *viewport* units, which are relative to the viewport used to draw the entire document. Similar to the media queries discussed in [Chapter 3](#), the size of the document region for images and embedded objects is different than for inline SVG. Viewport units also may not be well supported in SVG.

vw

Viewport width units. 1% of the viewport width.

vh

Viewport height units. 1% of the viewport height.

vmin

Viewport minimum units. 1% of the viewport width or height, whichever is smaller.

vmax

Viewport maximum units. 1% of the viewport width or height, whichever is larger.

Percentage Lengths

In SVG, most percentages are calculated relative to the height and width of the nearest ancestor element that defines a coordinate system. As described in [Chapter 8](#), the height and width refer to the scaled `viewBox` graphic region, not the available space.

Percentages for width or x-position are relative to the coordinate system width, percentages for height or y-position are relative to the coordinate system height, and percentages for all other properties are calculated as follows:

$$\text{distance} = \frac{\text{percentage value}}{100\%} \times \frac{\sqrt{(\text{width})^2 + (\text{height})^2}}{\sqrt{2}}$$

Angle Units

In SVG 1, there were no angle units. Angles were always specified only as numbers, and implicitly meant degrees. The angles in CSS `hsl()` color notation work the same way. (In contrast, in JavaScript trigonometry, angles are always implicitly radians.)

CSS Values and Units Level 3 adds angle units. They are used in the CSS transform syntax and CSS linear gradients. SVG 2 adopts them for other angle-related properties and attributes—but for best support, continue to use unitless degrees for most SVG attributes.

Positive angles are normally clockwise in both CSS and SVG (unless a negative scale has created a mirror reflection).

In most CSS contexts, 0° is straight up. In most SVG contexts, 0° is pointing right, towards the positive x-axis.

You can usually use values less than 0 or greater than a full circle. When animating and transitioning angle values, this allows you to control the direction of transition and the number of times something rotates.

deg

Degrees. We use the symbol version (°) in the book text.

- 360deg is a full-circle turn
- 90deg is a right-angle

rad

Radians.

- $2\pi \cdot 1\text{rad}$ (approximately 6.283rad) is a full-circle turn
- $(\pi/2) \cdot 1\text{rad}$ (approximately 1.571rad) is a right-angle
- 1rad is the angle such that a circular arc has the same path length as its radius.

grad

Gradians. (A rarely-used metric degree unit.)

- 400grad is a full-circle turn
- 100grad is a right-angle

turn

Turns.

- 1turn is a full-circle
- 0.25turn is a right-angle

Time Units

SVG 1 used time units from SMIL for the SVG/SMIL animation elements. CSS uses a more limited set of time units for animations and transitions.

Most JavaScript time-related methods use milliseconds (implicitly, as a number). However, the SVG/SMIL and CSS animation DOM methods use seconds.

The following units can be used in both CSS animations and SVG/SMIL animations:

s

Seconds.

ms

Milliseconds.

- 1000ms = 1s

The following units and formats are also valid in SVG/SMIL:

min

Minutes.

- 1min = 60s

h

Hours.

- 1h = 60min = 3600s

mm:ss

Minutes and seconds.

hh:mm:ss

Hours, minutes, and seconds.

- These are both implicit units based on the number format with :
- The minutes and seconds parts must always have 2 digits, between 00 and 59
- The seconds part can optionally have a decimal part, like **00:02.5** for 2.5s

In time-related SVG/SMIL attributes, numbers without units are treated as an amount in seconds.

This website is created and maintained by Amelia Bellamy-Royds, using material created by all three authors of the book.

View the [GitHub repository](#) to download the files, suggest a correction, or see when a page was last updated.