Units in CSS

Mohamed Saeed Ismail Salama
25-12064
T-9

CSS has several different units for expressing length. Which is used in many CSS properties and attributes, such as width, margin, padding, font-size, border-width, etc.

Length is expressed using a number followed by a length unit, such as 12px, 14pt, etc.

A whitespace cannot appear between the number and the unit. However, if the value is 0, the unit can be omitted.

For some CSS properties, negative lengths are allowed.

There are two types of length units: relative and absolute.

Some have their history in typography, such as point (pt) and pica (pc), others are known from everyday use, such as centimeter (cm) and inch (in). And there is also a “magic” unit that was invented specifically for CSS: the px.Does that mean different properties need different units?

No, the units have nothing to do with the properties, but everything with the output media: screen or paper.

There is no restriction on which units can be used where. If a property accepts a value in px (margin: 5px) it also accepts a value in inches or centimeters (margin: 1.2in; margin: 0.5cm) and vice-versa.

The so-called absolute units (cm, mm, in, pt and pc) mean the same in CSS as everywhere else, but only if your output device has a high enough resolution. On a laser printer, 1cm should be exactly 1 centimeter. But on low-resolution devices, such as computer screens, CSS doesn't require that. And indeed, the result tends to be different from one device to another and from one CSS implementation to another. It's better to reserve these units for high-resolution devices and in particular for printed output. On computer screens and handheld devices, you'll probably not get what you expect.

There is another reason to avoid absolute units for other uses than print: You look at different screens from different distances. 1cm on a desktop screen looks small. But the same on a mobile phone directly in front of your eyes looks big. It's better to use relative units, such as em, instead.

The em and ex units depend on the font and may be different for each element in the document. The em is simply the font size. In an element with a 2in font, 1em thus means 2in. Expressing sizes, such as margins and paddings, in em means they are related to the font size, and if the user has a big font (e.g., on a big screen) or a small font (e.g., on a handheld device), the sizes will be in proportion. Declarations such as text-indent: 1.5em and margin: 1em are extremely common in CSS.

The ex unit is rarely used. Its purpose is to express sizes that must be related to the x-height of a font. The x-height is, roughly, the height of lowercase letters such as a, c, m, or o. Fonts that have the same size (and thus the same em) may vary wildly in the size of their lowercase letters, and when it is important that some image, e.g., matches the x-height, the ex unit is available.

The px unit is the magic unit of CSS. It is not related to the current font and usually not relatated to physical centimeters or inches either. The px unit is defined to be small but visible, and such that a horizontal 1px wide line can be displayed with sharp edges (no anti-aliasing). What is sharp, small and visible depends on the device and the way it is used: do you hold it close to your eyes, like a mobile phone, at arms length, like a computer monitor, or somewhere in between, like an e-book reader? The px is thus not defined as a constant length, but as something that depends on the type of device and its typical use.

CSS also defines that raster images (such as photos) are, by default, displayed with one image pixel mapping to 1px. A photo with a 600 by 400 resolution will be 600px wide and 400px high. The pixels in the photo thus do not map to pixels of the display device (which may be very small), but map to px units. That makes it possible to exactly align images to other elements of a document, as long as you use px units in your style sheet, and not pt, cm, etc.

The magic unit of CSS, the px, is a often a good unit to use, especially if the style requires alignment of text to images, or simply because anything that is 1px wide or a multiple of 1px is guaranteed to look sharp.

But for font sizes it is even better to use em. The idea is (1) to not set the font size of the BODY element (in HTML), but use the default size of the device, because that is a size that the reader can comfortably read; and (2) express font sizes of other elements in em: H1 {font-size: 2.5em} to make the H1 2½ times as big as the normal, body font.

The only place where you could use pt (or cm or in) for setting a font size is in style sheets for print, if you need to be sure the printed font is exactly a certain size. But even there using the default font size is usually better.

To make it even easier to write style rules that depend only on the default font size, CSS has since 2013 a new unit: the rem. The rem (for “root em”) is the font size of the root element of the document. Unlike the em, which may be different for each element, the rem is constant throughout the document. E.g., to give P and H1 elements the same left margin, compare this pre-2013 style sheet:

```
p { margin-left: 1em }
h1 { font-size: 3em; margin-left: 0.333em }
```

with the new version:

```
p { margin-left: 1rem }
h1 { font-size: 3em; margin-left: 1rem }
```

Other new units make it possible to specify sizes relative to the reader's window. These are the vw and vh. The vw is 1/100th of the window's width and the vh is 1/100th of the window's height. There is also vmin, which stands for whichever is the smallest of vw and vh. And vmax. (You can guess what it does.)

Because they are so new, they don't work everywhere yet. But, as of early 2015, several browsers support them.

Compatibility

Unit	Chrome	IE	Firefox	Safari	Opera
em, ex, %, px, cm, mm, in, pt, pc	1.0	3.0	1.0	1.0	3.0
ch	27.0	9.0	1.0	7.0	20.0
rem	4.0	9.0	3.0	4.0	11.0
vh, vw	20.0	9.0	19.0	6.0	20.0
vmin	20.0	9.0	19.0	6.0	20.0
vmax	26.0	Not supported	19.0	Not supported	20.0

Summary

Relative Length

Relative length units specify a length relative to another length property. Relative length units scales better between different rendering mediums.

- em: relative to the font-size of the element (2em means 2 times the size of the current font)
- ex: relative to the x-height of the current font (rarely used)
- ch: relative to width of the "0" (zero)
- rem: relative to font-size of the root element
- vw: relative to 1% of the width of the viewport*
- vh: relative to 1% of the height of the viewport*
- vmin: relative to 1% of viewport's* smaller dimension
- vmax: relative to 1% of viewport's* larger dimension

* Viewport = the browser window size. If the viewport is 50cm wide, 1vw = 0.5cm.

Absolute Length

The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

Absolute length units are not recommended for use on screen, because screen sizes vary so much. However, they can be used if the output medium is known, such as for print layout.

- cm: centimeters
- mm: millimeters
- in: inches (1in = 96px = 2.54cm)
- px: pixels (1px = 1/96th of 1in)
- pt: points (1pt = 1/72 of 1in)
- pc: picas (1pc = 12 pt)