Should I use pt or px?

Asked 9 years, 3 months ago Active 3 years, 10 months ago Viewed 128k times

147

What is the difference between `pt` and `px` in CSS? Which one should I use and why?

css

44

edited Jan 26 '14 at 9:26
sawa
148k 31 226 324

asked Aug 24 '10 at 14:03

esqew
28.9k 21 79 112

5 Answers

62

`px` ≠ Pixels

All of these answers seem to be incorrect. Contrary to intuition, in CSS the `px` **is not pixels**. At least, not in the simple physical sense.

Read this article from the W3C, EM, PX, PT, CM, IN…, about how `px` is a "magical" unit invented for CSS. The meaning of `px` varies by hardware and resolution. (That article is fresh, last updated 2014-10.)

My own way of thinking about it: `1 px is the size of a thin line intended by a designer to be barely visible.`

To quote that article:

> The px unit is the magic unit of CSS. It is not related to the current font and also not related to the absolute units. 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 a book? The px is thus not defined as a constant length, but as something that depends on the type of device and its typical use.

> Nowadays there are devices that could in principle display smaller sharp dots (although you might need a magnifier to see them). But documents from the last century that used px in CSS still look the same, no matter what the device. Printers, especially, can display sharp lines with much smaller details than 1px, but even on printers, a 1px line looks very much the same as it would look on a computer monitor. Devices change, but the px always has the same visual appearance.

That article gives some guidance about using `pt` vs `px` vs `em`, to answer this Question.

edited Dec 19 '14 at 4:11

answered Dec 19 '14 at 4:06

Basil Bourque

149k 39 477 669

So, can I use `px` in css like `pt` on iOS and `dp` on Android? – GRiMe2D Apr 16 '16 at 9:47

4 This is the only correct answer. The quoted w3.org url is updated from http to https as: https://www.w3.org/Style/Examples/007/units.en.html – hyyou2010 Jan 2 '17 at 15:29

4 thin line intended by a designer to be barely visible. WAT? This is a demagogy. Pixel in css == real pixel * device pixel ratio. No less, no more. In fact, it does not matters in terms of current question and work with css. – extempl Aug 9 '17 at 7:33

"(...) and such that a horizontal 1px wide line can be displayed with sharp edges (no anti-aliasing):" FALSE. Open a 1px solid line in the native Android browser (not Chrome), and the line will be horribly blurred on hdpi devices depending on its location on screen. – andreszs Aug 27 '17 at 0:12

58

Here you've got a very detailed explanation of their differences

http://kyleschaeffer.com/development/css-font-size-em-vs-px-vs-pt-vs/

The jist of it (from source)

> **Pixels** are fixed-size units that are used in screen media (i.e. to be read on the computer screen). Pixel stands for "picture element" and as you know, one pixel is one little "square" on your screen. **Points** are traditionally used in print media (anything that is to be printed on paper, etc.). One point is equal to 1/72 of an inch. Points are much like pixels, in that they are fixed-size units and cannot scale in size.

edited Feb 1 '16 at 8:34

Mathias

answered Aug 24 '10 at 14:05

Flakron Bytyqi

11 I like that site but it doesn't really explain what the % is of. 100% of what? My car? – Joe Phillips Aug 24 '10 at 14:49

7 @Joe Phillips - 100% will be the font-size you set on the last parent element. If you haven't set any font-size anywhere, 100% is the default browser font-size (which is 16px by default in most browsers - can be changed trough browser options) – easwee Aug 24 '10 at 15:32

3 One px is 1/96 of an in. On most *desktop* monitors it will correspond to one device pixel. But on other devices, all units may be scaled so that a px is not the same as a device pixel. Other absolute units however will be continued to scale equally so one "in" will still be 96 "px". – thomasrutter Nov 12 '12 at 2:28

59 **INCORRECT** Actually, that article at kyleschaeffer is incorrect. The `px` is *not* pixels in CSS, at least not in the simple physical sense. Read this article by the WC3 explaining that `px` is a "magic" unit invented solely by and for CSS. The meaning of `px` varies by hardware and screen resolution. – Basil Bourque Dec 19 '14 at 3:56

5 downvoting this answer to the prehistoric cave it belongs to. Please edit it – Toskan Apr 20 '17 at 21:45

51

Have a look at this excellent article at CSS-Tricks:

- px – em – % – pt – keyword

Taken from the article:

pt

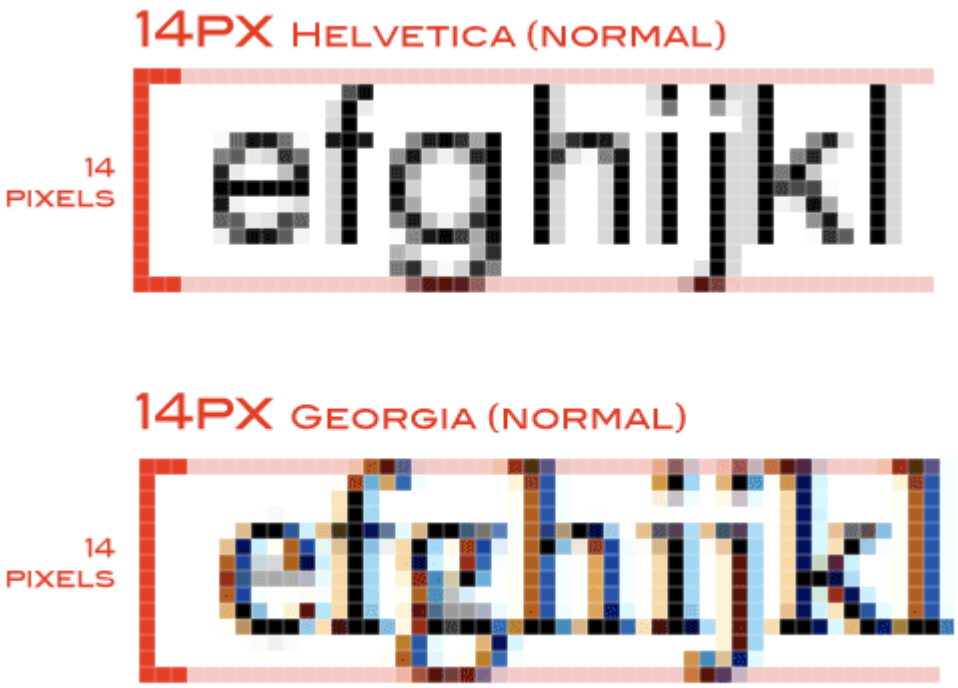
The final unit of measurement that it is possible to declare font sizes in is point values (pt). **Point values are only for print CSS!** A point is a unit of measurement used for real-life ink-on-paper typography. 72pts = one inch. One inch = one real-life inch like-on-a-ruler. Not an inch on a screen, which is totally arbitrary based on resolution.

Just like how pixels are dead-accurate on monitors for font-sizing, point sizes are dead-accurate on paper. For the best cross-browser and cross-platform results while printing pages, set up a print stylesheet and size all fonts with point sizes.

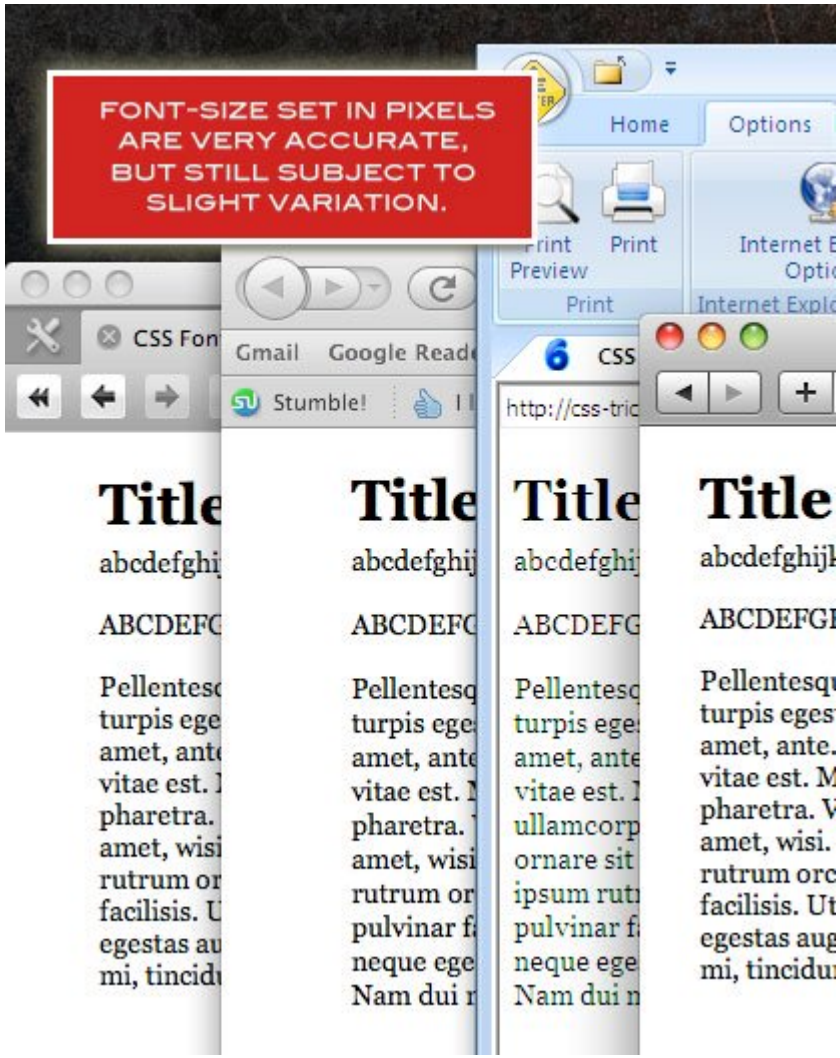
For good measure, the reason we don't use point sizes for screen display (other than it being absurd), is that the cross-browser results are drastically different:

px

If you need fine-grained control, sizing fonts in pixel values (px) is an excellent choice (it's my favorite). On a computer screen, it doesn't get any more accurate than a single pixel. With sizing fonts in pixels, you are literally telling browsers to render the letters exactly that number of pixels in height:



Windows, Mac, aliased, anti-aliased, cross-browsers, doesn't matter, a font set at 14px will be 14px tall. But that isn't to say there won't still be some variation. In a quick test below, the results were slightly more consistent than with keywords but not identical:



Due to the nature of pixel values, they do not cascade. If a parent element has an 18px pixel size and the child is 16px, the child will be 16px. However, font-sizing settings can be using in combination. For example, if the parent was set to 16px and the child was set to larger, the child would indeed come out larger than the parent. A quick test showed me this:

"Larger" bumped the 16px of the parent into 20px, a 25% increase.

Pixels have gotten a bad wrap in the past for accessibility and usability concerns. In IE 6 and below, font-sizes set in pixels *cannot be resized by the user*. That means that us hip young healthy designers can set type in 12px and read it on the screen just fine, but when folks a little longer in the tooth go to bump up the size so they can read it, they are unable to. This is really IE 6's fault, not ours, but we gots what we gots and we have to deal with it.

Setting font-size in pixels is the most accurate (and I find the most satisfying) method, but do take into consideration the number of visitors still using IE 6 on your site and their accessibility needs. We are right on the bleeding edge of not needing to care about this anymore.

edited Feb 8 '17 at 14:30
Community ♦
1 1

answered Aug 24 '10 at 14:05
Sarfraz
331k 67 490 553

2 For the question(above), This is the answer – overexchange Sep 13 '16 at 3:55

7 **INCORRECT** The `px` is *not* pixels in CSS, at least not in the simple physical sense. Read this article by the WC3 explaining that `px` is a "magic" unit invented solely by and for CSS. The meaning of `px` varies by hardware and screen resolution. – Basil Bourque Jul 5 '17 at 19:24

3

pt is a derivation (abbreviation) of "point" which historically was used in print type faces where the size was commonly "measured" in "points" where 1 point has an approximate measurement of 1/72 of an inch, and thus a 72 point font would be 1 inch in size.

px is an abbreviation for "pixel" which is a simple "dot" on either a screen or a dot matrix printer or other printer or device which renders in a dot fashion - as opposed to old typewriters which had a fixed size, solid striker which left an imprint of the character by pressing on a ribbon, thus leaving an image of a fixed size.

Closely related to point are the terms "uppercase" and "lowercase" which historically had to do with the selection of the fixed typographical characters where the "captital" characters where placed in a box (case) above the non-captitalized characters which were place in a box below, and thus the "lower" case.

There were different boxes (cases) for different typographical fonts and sizes, but still and "upper" and "lower" case for each of those.

Another term is the "pica" which is a measure of one character in the font, thus a pica is 1/6 of an inch or 12 point units of measure (12/72) of measure.

Strickly speaking the measurement is on computers 4.233mm or 0.166in whereas the old point (American) is 1/72.27 of an inch and French is 4.512mm (0.177in.). Thus my statement of "approximate" regarding the measurements.

Further, typewriters as used in offices, had either and "Elite" or a "Pica" size where the size was 10 and 12 characters per inch repectivly.

Additionally, the "point", prior to standardization was based on the metal typographers "foot" size, the size of the basic footprint of one character, and varied somewhat in size.

Note that a typographical "foot" was originally from a deceased printers actual foot. A typographic foot contains 72 picas or 864 points.

As to CSS use, I prefer to use EM rather than px or pt, thus gaining the advantage of scaling without loss of relative location and size.

EDIT: Just for completeness you can think of EM (em) as an element of measure of one font height, thus 1em for a 12pt font would be the height of that font and 2em would be twice that height. Note that for a 12px font, 2em is 24 pixels. SO 10px is typically 0.63em of a standard font as "most" browsers base on 16px = 1em as a standard font size.

edited Aug 24 '10 at 14:35

answered Aug 24 '10 at 14:26

Mark Schultheiss

26.5k 8 57 85

17

A px is a pixel, which will map on to a screen pixel in most cases.

CSS provides a bunch of other units, and which one you should choose depends on what you are setting the size of.

A pixel is great if you need to size something to match an image, or if you want a thin border.

Percentages are great for font sizes as, if you use them consistently, you get font sizes proportional to the user's preference.

Ems are great when you want an element to size itself based on the font size (so a paragraph might get wider if the font size is larger)

… and so on.

answered Aug 24 '10 at 14:05

Quentin

701k 83 953 1101

1 Hence "in most cases". – Quentin Aug 24 '10 at 14:10

@BasilBourque — Hence "in most cases" – Quentin Jul 5 '17 at 21:16

@BasilBourque — Within the context of CSS, the cases vary by display type. Most are 1:1 CSS pixel to screen pixel. (Although a little less so these days given ultra high density displays) – Quentin Jul 6 '17 at 6:56

Highly active question. Earn 10 reputation in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.