**Lengths**

You have already seen that the values of some CSS properties are given as lengths (such as the size of fonts, height of lines of text, and gaps between words and letters), and you will come across more properties whose values are expressed as lengths later in the chapter.

So, let’ s take a moment to look at the **three ways lengths** can be specified in CSS:

* Relative units
* Absolute units
* Percentages

**Relative Units:**

There are three types of relative units: pixels, which relate to the resolution of the screen, and em’ s and ex’ s both of which relate to the size of fonts.

***px:***

The pixel, referred to in code as px, is by far the most commonly used unit of length in CSS. A pixel is the smallest unit of resolution on a screen and if you look very closely at your screen you might just be able to see the square dots that are the pixels.

Technically, the size of a layout that uses pixels as a unit of measurement can depend upon the viewing medium (keep reading to see why I say “can”), which is why it is counted as a relative unit.

Most computer screens have a resolution of 72 dots per inch (dpi), but you will find that laser and bubble jet printers are usually set with a higher resolution — my current printer runs at 300 dpi. In contrast, mobile devices can have a lower resolution than computer screens or (in the case of some smart phones such as the iPhone) a higher resolution.

So, a table that is 500 pixels wide could be 9.9444 inches wide on a 72 dpi screen, 1.666 inches wide at 300 dpi, or 13.888 inches wide on a 32 dpi screen (and a screen that is only 32 dpi is unlikely to be that much over 13 inches wide).

In reality, when you print a web page from your browser, it will adjust the pixels to present a readable version of the document. In fact, CSS recommends that in such cases user agents rescale pixel units so that reading at arm’ s length 1 pixel would correspond to about 0.28 mm or 1/90 of an inch.

***em:***

An em is equivalent to the height of the current font, and because the size of fonts can vary throughout a document, the height of the em unit can be different in different parts of the document. Furthermore, because users can change the size of text in their browser, the em unit is capable of varying in relation to the size of the text that the user has selected.

This means that the em unit is most commonly used for measurements of elements that contain text and for controlling spacing between text (for example it can be used in the line - height property to set the gaps between lines of text in relation to their height).

While the em unit is equivalent to the height of a font, ***it is often thought to have derived from the width of a lowercase m***; you may also hear the term *en*, which equates to half an *em*.

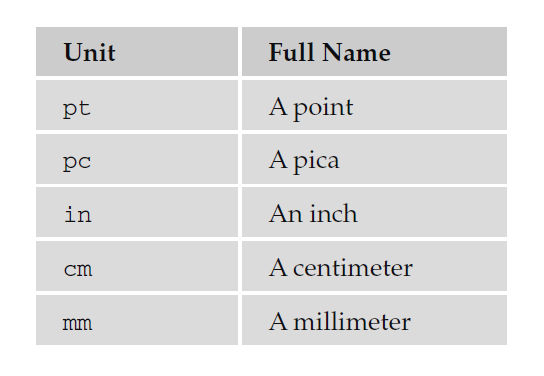
***ex:***

The ex should be the height of a lowercase x. Because different fonts have different proportions, the ex is related to the font size and the type of font. In Figure below, you can see the x in the Courier typeface is smaller than the x in the Impact typeface.



**Absolute Units:**

Generally speaking, absolute units are used far less than relative units (and you will rarely come across designs that use these units of measurement). The following table shows the absolute units that are used in some CSS properties.



I should’ t really needs to clarify inches, millimeters, or centimeters, but the other two are more interesting. A point is 1/72 of an inch (the same as a pixel in most computer screen resolutions), and a pica is 1/12 of an inch (12 points). Typographers tend to use points to measure font sizes and **leading** (the gaps between lines), while picas are used to measure line lengths.

**Percentages:**

Percentages give a value in relation to another value. For example, if your page only contained two paragraphs, and you wanted each to take up half of the width of the browser, then the paragraphs might be given a width property with a value of 50%. However, if the <p> element were inside another element that you knew was 500 pixels wide, they would take up 50 percent of the width of that containing element (or 250 pixels) each.