4.3.2 Lengths

Lengths refer to distance measurements.

The format of a length value (denoted by <length> in this specification) is a [<number>](http://www.w3.org/TR/CSS21/syndata.html#value-def-number) (with or without a decimal point) immediately followed by a unit identifier (e.g., px, em, etc.). After a zero length, the unit identifier is optional.

Some properties allow negative length values, but this may complicate the formatting model and there may be implementation-specific limits. If a negative length value cannot be supported, it should be converted to the nearest value that can be supported.

If a negative length value is set on a property that does not allow negative length values, the declaration is ignored.

In cases where the [used](http://www.w3.org/TR/CSS21/cascade.html#usedValue) length cannot be supported, user agents must approximate it in the [actual value.](http://www.w3.org/TR/CSS21/cascade.html#actual-value)

There are two types of length units: relative and absolute. *Relative length* units specify a length relative to another length property. Style sheets that use relative units can more easily scale from one output environment to another.

Relative units are:

* **em**: the ['font-size'](http://www.w3.org/TR/CSS21/fonts.html#propdef-font-size) of the relevant font
* **ex**: the 'x-height' of the relevant font

h1 { margin: 0.5em } /\* em \*/

h1 { margin: 1ex } /\* ex \*/

The 'em' unit is equal to the computed value of the ['font-size'](http://www.w3.org/TR/CSS21/fonts.html#propdef-font-size) property of the element on which it is used. The exception is when 'em' occurs in the value of the 'font-size' property itself, in which case it refers to the font size of the parent element. It may be used for vertical or horizontal measurement. (This unit is also sometimes called the quad-width in typographic texts.)

The 'ex' unit is defined by the element's first available font. The exception is when 'ex' occurs in the value of the ['font-size'](http://www.w3.org/TR/CSS21/fonts.html#propdef-font-size) property, in which case it refers to the 'ex' of the parent element.

The 'x-height' is so called because it is often equal to the height of the lowercase "x". However, an 'ex' is defined even for fonts that do not contain an "x".

The x-height of a font can be found in different ways. Some fonts contain reliable metrics for the x-height. If reliable font metrics are not available, UAs may determine the x-height from the height of a lowercase glyph. One possible heuristic is to look at how far the glyph for the lowercase "o" extends below the baseline, and subtract that value from the top of its bounding box. In the cases where it is impossible or impractical to determine the x-height, a value of 0.5em should be used.

The rule:

h1 { line-height: 1.2em }

means that the line height of "h1" elements will be 20% greater than the font size of the "h1" elements. On the other hand:

h1 { font-size: 1.2em }

means that the font-size of "h1" elements will be 20% greater than the font size inherited by "h1" elements.

When specified for the root of the [document tree](http://www.w3.org/TR/CSS21/conform.html#doctree) (e.g., "HTML" in HTML), 'em' and 'ex' refer to the property's [initial value](http://www.w3.org/TR/CSS21/about.html#initial-value).

Child elements do not inherit the relative values specified for their parent; they inherit the [computed values](http://www.w3.org/TR/CSS21/cascade.html#computed-value).

In the following rules, the computed ['text-indent'](http://www.w3.org/TR/CSS21/text.html#propdef-text-indent) value of "h1" elements will be 36px, not 45px, if "h1" is a child of the "body" element.

body {

font-size: 12px;

text-indent: 3em; /\* i.e., 36px \*/

}

h1 { font-size: 15px }

*Absolute length* units are fixed in relation to each other. They are mainly useful when the output environment is known. The absolute units consist of the physical units (in, cm, mm, pt, pc) and the px unit:

* **in**: inches — 1in is equal to 2.54cm.
* **cm**: centimeters
* **mm**: millimeters
* **pt**: points — the points used by CSS are equal to 1/72nd of 1in.
* **pc**: picas — 1pc is equal to 12pt.
* **px**: pixel units — 1px is equal to 0.75pt.

For a CSS device, these dimensions are either anchored (i) by relating the physical units to their physical measurements, or (ii) by relating the pixel unit to the *reference pixel*. For print media and similar high-resolution devices, the anchor unit should be one of the standard physical units (inches, centimeters, etc). For lower-resolution devices, and devices with unusual viewing distances, it is recommended instead that the anchor unit be the pixel unit. For such devices it is recommended that the pixel unit refer to the whole number of device pixels that best approximates the reference pixel.

Note that if the anchor unit is the pixel unit, the physical units might not match their physical measurements. Alternatively if the anchor unit is a physical unit, the pixel unit might not map to a whole number of device pixels.

Note that this definition of the pixel unit and the physical units differs from previous versions of CSS. In particular, in previous versions of CSS the pixel unit and the physical units were not related by a fixed ratio: the physical units were always tied to their physical measurements while the pixel unit would vary to most closely match the reference pixel. (This change was made because too much existing content relies on the assumption of 96dpi, and breaking that assumption breaks the content.)

The *reference pixel* is the visual angle of one pixel on a device with a pixel density of 96dpi and a distance from the reader of an arm's length. For a nominal arm's length of 28 inches, the visual angle is therefore about 0.0213 degrees. For reading at arm's length, 1px thus corresponds to about 0.26 mm (1/96 inch).

The image below illustrates the effect of viewing distance on the size of a reference pixel: a reading distance of 71 cm (28 inches) results in a reference pixel of 0.26 mm, while a reading distance of 3.5 m (12 feet) results in a reference pixel of 1.3 mm.

Showing that pixels must become
larger if the viewing distance increases   [[D]](http://www.w3.org/TR/CSS21/images/longdesc/pixel1-desc.html" \o "Long description for the illustration of a pixel's definition)

This second image illustrates the effect of a device's resolution on the pixel unit: an area of 1px by 1px is covered by a single dot in a low-resolution device (e.g. a typical computer display), while the same area is covered by 16 dots in a higher resolution device (such as a printer).

Showing that more device pixels (dots)
are needed to cover a 1px by 1px area on a high-resolution device than
on a low-res one   [[D]](http://www.w3.org/TR/CSS21/images/longdesc/pixel2-desc.html" \o )

h1 { margin: 0.5in } /\* inches \*/

h2 { line-height: 3cm } /\* centimeters \*/

h3 { word-spacing: 4mm } /\* millimeters \*/

h4 { font-size: 12pt } /\* points \*/

h4 { font-size: 1pc } /\* picas \*/

p { font-size: 12px } /\* px \*/