

CSS Units

Mohamed Emary

October 20, 2023

1 CSS Units

CSS has multiple units for sizing elements. Some are relative, some are absolute and some are based on the viewport.

1.1 Absolute Units

Absolute units are those that are always the same in any context. `px` is the only absolute unit you should be using for web projects. Other absolute units are more suited for print setting because they are related to physical units such as `in` (inch) and `cm` (centimeter).

Unit	Name	Equivalent to
<code>cm</code>	Centimeters	<code>1cm = 37.8px = 25.2/64in</code>
<code>mm</code>	Millimeters	<code>1mm = 1/10th of 1cm</code>
<code>Q</code>	Quarter-millimeters	<code>1Q = 1/40th of 1cm</code>
<code>in</code>	Inches	<code>1in = 2.54cm = 96px</code>
<code>pc</code>	Picas	<code>1pc = 1/6th of 1in</code>
<code>pt</code>	Points	<code>1pt = 1/72nd of 1in</code>
<code>px</code>	Pixels	<code>1px = 1/96th of 1in</code>

Most of these units are more useful when used for print, rather than screen output.

1.2 Relative units

Relative units are units that are relative to something else, perhaps the size of the parent element's font, or the size of the viewport. `em` and `rem` both refer to a font size, though they are often used to define other sizes in CSS. You'll see both of them often so we're going to explain both, but **as a rule-of-thumb, prefer `rem`**.

Unit	Relative to
<code>em</code>	Font size of the parent, in the case of typographical properties like <code>font-size</code> , and font size of the element itself, in the case of other properties like <code>width</code> .
<code>ex</code>	x-height of the element's font.
<code>ch</code>	The advance measure (width) of the glyph "0" of the element's font.
<code>rem</code>	Font size of the root element.
<code>lh</code>	Line height of the element.
<code>rlh</code>	Line height of the root element. When used on the <code>font-size</code> or <code>line-height</code> properties of the root element, it refers to the properties' initial value.
<code>vw</code>	Percentage of the viewport's width.
<code>vh</code>	Percentage of the viewport's height.
<code>vmin</code>	Percentage of the viewport's smaller dimension.
<code>vmax</code>	Percentage of the viewport's larger dimension.
<code>vb</code>	Percentage of the size of the initial containing block in the direction of the root element's block axis.
<code>vi</code>	Percentage of the size of the initial containing block in the direction of the root element's inline axis.

Unit	Relative to
svw, svh	Percentage of the small viewport's width and height, respectively.
lvw, lvh	Percentage of the large viewport's width and height, respectively.
dvw, dvh	Percentage of the dynamic viewport's width and height, respectively.

1em is the **font-size** of an **element** (or the **element's parent** if you're using it to set **font-size**). So, for example, if an element's **font-size** is **16px**, then setting its width to **4em** would make its width **64px** ($16 * 4 == 64$).

1rem is the **font-size** of the **root element** (either **:root** or **html**). The math works the same with **rem** as it did with **em**, but without the added complexity of keeping track of the parent's font size. Relying on **em** could mean that a particular size could change if the context changes, which is very likely not the behavior you want.

1.3 Viewport Units

The units **vh** and **vw** relate to the size of the viewport. Specifically, **1vh** is equal to **1%** of the viewport height and **1vw** is equal to **1%** of the viewport width. These can be useful any time you want something to be sized relative to the viewport, examples including full-height heroes, full-screen app-like interfaces.

2 Important note for me

Read the 3 articles at the assignment section of this lesson again with more attention.