

Theory: Relative units

🕒 12 minutes 0 / 5 problems solved

Skip this topic

Start practicing

823 users solved this topic. Latest completion was about 2 hours ago.

At this point, you are probably comfortable working with pixels, but CSS supports many other units of measurement, for example, percentages. Any of these units could be used depending on your goals and preferences. In CSS, it is common to distinguish two major types of units: relative and absolute. **Relative units of measurement** describe property values that directly depend on the values of the same properties for the parent element. This is what we are going to talk about in this topic.

§1. Relative units

Let's look at several relative units of measurement and see where they differ.

- `%` sets the measurement as a percentage of the parent element. This is especially convenient when you need to make a page that adjusts to the size of the screen where it is viewed. In CSS, percentages are calculated from the size of the desired parent element property. With the help of percentages, it is convenient to set the size of block elements. In the example below, the height of the `<p>` element will be 200 pixels, because it is exactly 50% of the height of the parent element `<div>` :

```
1 <div>
2   <p>Units of measure</p>
3 </div>
```

```
1 div {
2   height: 400px;
3 }
4
5 p {
6   height: 50%; /* 200px */
7 }
```

- `em` is used to indicate the font size relative to the current font of the parent element. If the font size is `4.5em`, it means that it is 4.5 times larger than the current font of the parent element:

```
1 <p><b>Units</b> of measure</p>
```

```
1 p {
2   font-size: 10px;
3 }
4
5 b {
6   font-size: 2em; /* 20px */
7 }
```

- `rem` resembles `em`, but with one important difference: `rem` is used if you want to change the font size of the element in relation to the font size of the root element `<html>` :

Current topic:

Relative units Stage 2 ...

Topic depends on:

✗ Comments Stage 2 ...

Topic is required for:

Width and height Stage 2 ...

Margin and padding Stage 2 ...

Background Stage 2 ...

Box-Shadow ...

line-height ...

Introduction to animation ...

Position Stage 2 ...

Positioning Properties Stage 2 ...

Table of contents:

[↑ Relative units](#)

[§1. Relative units](#)

[§2. Viewport units](#)

[§3. Rarely used units of measurement](#)

[§4. Specifying unit values](#)

[Feedback & Comments](#)

```
1  html {
2      font-size: 10px;
3  }
4
5  p {
6      font-size: 15px;
7  }
8
9  b {
10     font-size: 2rem; /* 20px */
11 }
12
```

Note that the text font size in the `` tag will be 20 pixels, even though the styles are applied to the `<p>` tag.

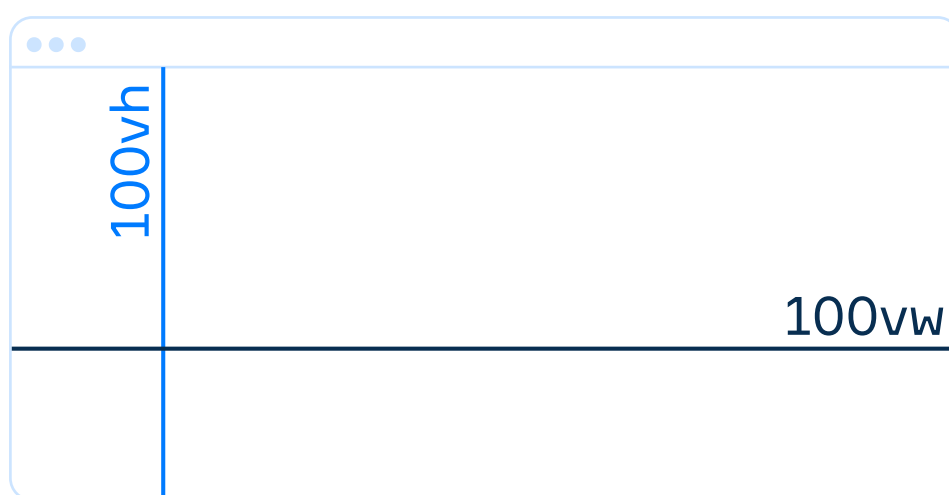
Be careful: `em` is based on the font size of the parent element, while `rem` depends on the `<html>` element. Do not confuse the two! The `<html>` itself is also parent to some other elements, which can cause some confusion.

So, how do you set the font size with `%`, `em`, and `rem` if the original font size of the `<html>` element is not set? In browsers, the font size is set to **16px** by default. You can use this value in your calculations. In such cases, **16px** is equal to **100%**, **1em**, or **1rem**.

§2. Viewport units

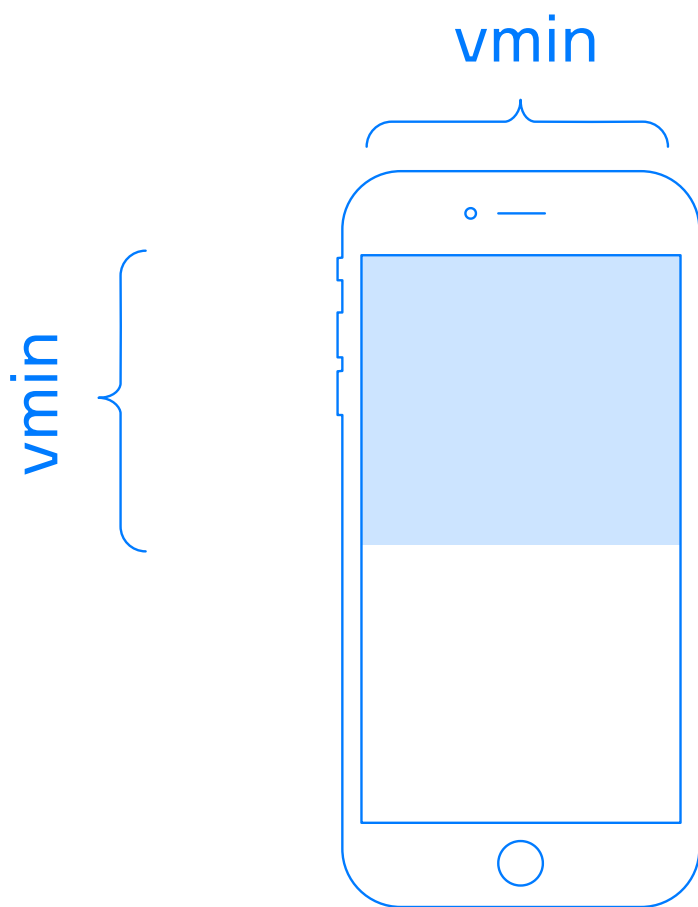
There are also relative units of measurement that depend on the viewport. The word "viewport" refers to the area of the web page that the user can see without having to scroll. Such units begin with the letter "v", which stands for "viewport". As you may have guessed, such units of measurement are used to ensure that the web page is correctly displayed on devices with different screen sizes.

- `vw` — 1% of the width of the viewport.
- `vh` — 1% of the height of the viewport.

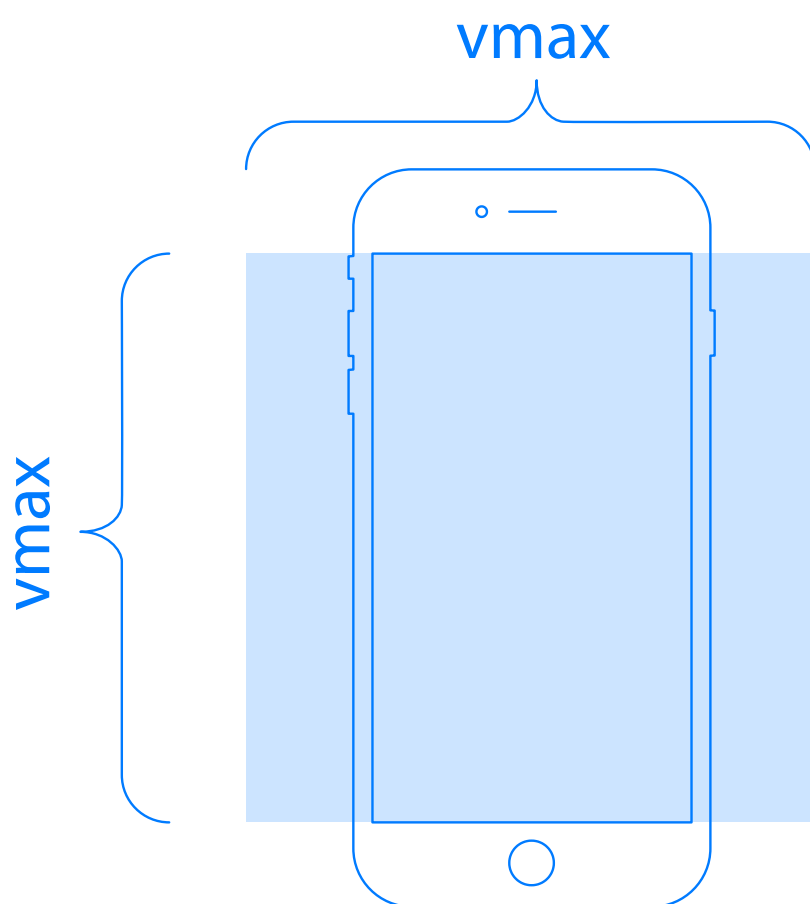


`100vw` and `100vh` are the values of the whole window together with the scroll bar. When you try to set this value, the elements will see a horizontal/vertical scroll bar.

- `vmin` — 1% of the minimum height and width of the viewport. `vmin` uses the size of the smaller side. For example, if the height of the browser window is less than the width, `1vmin` will be equal to `1vh`. If the browser width is less than its height, `1vmin` equals `1vw`.



- `vmax` — 1% of the maximum height and width of the viewport area. This unit of measurement is the direct opposite of `vmin` and uses the larger side. For example, if the width of the browser is greater than its height, `1vmax` is `1vw`. If the height of the browser is greater than its width, `1vmax` equals `1vh`.



`vmin` and `vmax` will be different when you resize your browser window or change your smartphone orientation.

The viewport dependent measuring units, like `%`, are suitable for dimensioning block elements:

```
1 pre {  
2   width: 10vw;  
3   height: 24vh;  
4 }  
5  
6 div {  
7   width: 20vmin;  
8   height: 40vmax;  
9 }
```

§3. Rarely used units of measurement

Let's move on and consider a few units that are not used as often.

- `ex` allows you to set the font size relative to the x-height of the current font of the parent element. The term "x-height" implies the height of lower case letters. This unit is rarely used.

Here is an example of use:

```
1 p {  
2   font-size: 4ex;  
3 }
```

- `ch` is used to set the font size relative to the width of the character "O" in the current font of the parent element. This unit of measurement is especially good for printing. For example, if you are making a magazine, you might need to limit the sheet size to the number of characters. In mono-width fonts (fixed width fonts) where all characters have the same width, `1ch` is equal to one character. In such cases, `ch` may be especially useful. In proportional fonts (with variable width), any given character may be wider or narrower than the "O" character.

Here is an example:

```
1 p {  
2   font-size: 5ch;  
3 }
```

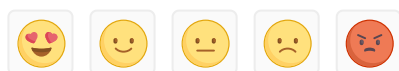
§4. Specifying unit values

Unit values can be specified as integers (`5vh`, `27%`, `5vw`) or decimals (`5.6em`, `1.2rem`, `25.5%`). If zero is used as a value, the indication of the unit of measurement can be omitted. The browser will understand the lines `font-size: 0%;` and `font-size: 0;` equally well. If a decimal begins with `0`, then `0` before the point can be omitted: `.9em`, `.5rem`.

Some properties in CSS allow you to specify negative values: `-35.5%`. We will learn about them later on. When specifying negative values, make sure they are acceptable for your properties. Otherwise, the browser will ignore the code line containing the error.

 Report a typo

73 users liked this theory. 2 didn't like it. What about you?



Start practicing

[Comments \(3\)](#)

[Hints \(0\)](#)

[Useful links \(0\)](#)

[Show discussion](#)