A Bite-Sized Best Practices Guide for CSS Units: Em, Rem, Px, and More

A beginner's guide to using CSS units like em, rem, px and more in your responsive web designs.

Celia Lewis Follow

Jun 24 · 3 min read ★

Absolute Units

Absolute units are divided mainly into two categories: pixels (px), and the rest (cm, mm, in). Absolute units are good for laying out precise screen dimensions, but they will not scale with the viewport, so they **aren't so great for creating responsive designs on tablets and mobile phones.**

You may recognize units such as cm, mm, and inches from your everyday life. These units, like their real-life counterparts, should only be utilized when making designs for print (physical designs). In general, avoid the non-pixel absolute units unless you have a specific use for them.

- **Pixels** (px)
- Also pt, cm, mm, in

⭐ **Best Practice**: While I advise that you **use the px unit sparingly if there is a better alternative**, they can sometimes be useful in **specifying specific widths, heights, and font sizes**.

Percentages

Percentages are primarily used for **widths**. They make it really easy to calculate what percent of a parent object's space a child element should occupy. For example, examine the HTML below, where "parent" is the parent container to two children:

```
<div class=”parent”>
   <div class=”child-one”></div>
   <div class=”child-two”></div>
</div>
```

By default, my parent div (the red block below) has a width of 100%. By declaring child-one to have a width of 60%, and child-two to have a width of 40%, the result will be:

Relative Units

These units are very commonly used to achieve responsive designs, since they scale with different elements of the page. This means they will look nice on monitors, tablets, and phones! **The em and the rem units should be your best friend as a responsive web designer.**

- Units relative to font-sizes. For example, **em**, **rem**, ch, and ex.

- Units relative to the viewport, like **vh**, **vw**, vmin, and vmax.

Things to know about Em and Rem

The Em Unit:

- When used on a **font-size** property, em is **relative to parent's font size.**
- **1em will change nothing**. It means copy the parent's font size.
- So 120% em is equivalent to 120% the font size of the parent.
- ⚠️ **A Problem**: **If you change the parent's font size, you will affect all subsequent font sizes for all child elements** of that parent. So by applying the em unit to children within a hierarchy, you could have some astronomical scaling effects if you change the root font size.
- ⭐ **Best Practice**: **Em is great for setting padding, margins, and widths**. This is because **when you use em for anything other than font-size, it is not relative to the parent.** It is relative to its own font-size.
- 👍 **So if you use the em unit on a width property, the width will scale with the font-size of that element**, not with the font-size of the parent element.

The Rem Unit:

- Instead of being relative to the parent, the **rem unit is relative to the root object**: the HTML element.
- You can **change the font size on the HTML element** to change the font-size on all objects using the rem unit.
- ⚠️ **A Problem**: Setting a font-size in the HTML element is **not required**. It can occasionally create problems down the road, so use it sparingly.
- ⭐ **Best Practice**: Rem units avoid the scaling problems of the Em unit, so they are a **great go-to when setting font-size for responsive web design.**

Responsive Design HTML CSS Web Development Programming