What are ems and rems in CSS, and when should you use them?

In the previous lecture, we learned about absolute length units like pixels. While absolute units can be helpful in certain situations, there will be times when you want to use relative units.

In this lecture, we will learn about two relative units: ems and rems.

em units are relative to the font size of the element. If you are using ems for the font-size property, the size of the text will be relative to the font size of the parent element.

To better understand how this works, let's take a look at an example:

<p class="para">I am a paragraph element</p>

<div class="blue-box"></div>

For the HTML, we have a paragraph and div element. The paragraph element has a class of para, and the div element has a class of blue-box.

Here is the accompanying CSS:

.para {

font-size: 20px;

margin-bottom: 1.5em;

border: 2px solid red;

}

.blue-box {

background-color: blue;

color: white;

padding: 10px;

}

For the para class, we set the font-size to 20px and the margin-bottom to 1.5em. This means that the margin will be 1.5 times the font size of the paragraph element. 1.5em results in 30 pixels of margin at the bottom of the paragraph. We have also set a border of 2px solid red so you can see the margins better.

For the blue-box class, we set the background color to blue, the text color to white, and the padding to 10px on all four sides.

From the example, there'll be a clear space between the bottom of the paragraph element and the blue box.

So what happens if we remove the font-size property from the para class?

Well, the bottom margin will be relative to the font size of the parent element. In this case, the parent element is the body element, which has a default font size of 16px.

Good use cases for ems would be when you are working with modular components like buttons or cards. By using em units, you can ensure that all aspects of the component (such as padding, margin, and borders) scale proportionally with the font size, keeping consistent proportions.

So, up until this point, we have been setting the font size for an element using pixels. But that does present some challenges for users.

Inside your browser settings, you can control the default font size.

For those with visual impairments, they may increase the font size to make it easier to read. But if you are setting pixels for the font sizes in your web designs, the text will not scale proportionally with the rest of the content.

One way to address this issue is to use rem units for typography. A rem unit is relative to the font size of the root element, which is the html element.

By default, the font size of the html element is 16px. If the user increases the font size in their browser settings, the font size of the html element will increase, and all rem units will scale proportionally.

Here is an example of using the rem unit for the font size instead of pixels:

.para {

font-size: 1.2rem;

margin-bottom: 1.5em;

border: 2px solid red;

}

By setting the font size to 1.2rem, the font size of the paragraph element will be 1.2 times the font size of the root element. If the user hasn't changed the default font size, the font size of the paragraph element will be 19.2px because it is 1.2 times 16px.

So when should you use rem units? rem units are preferred over pixels for typography because they scale proportionally with the user's browser settings. This makes your content more accessible to users with visual impairments.

rem units can also help maintain consistent spacing and layout across different elements.