CSS Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



1. **Content** - The content of the box, where text and images appear
2. **Padding** - Clears an area around the content. The padding is transparent
3. **Border** - A border that goes around the padding and content
4. **Margin** - Clears an area outside the border. The margin is transparent

The CSS Box Model is used to create a definition for the way the HTML elements are organized on the screen. This approach accounts for options such as margins, padding, borders, and all the properties that manipulate them. Each element can be thought of as having its own box.

The Difference Between PX, EM, REM, %, VW, and VH?

Absolute Units

**PX**: Pixels (px) are considered absolute units, although they are relative to the DPI and resolution of the viewing device. But on the device itself, the PX unit is fixed and does not change based on any other element. Using PX can be problematic for responsive sites, but they are useful for maintaining consistent sizing for some elements. If you have elements that should not be resized, then using PX is a good choice.

Relative Units

**EM**: Relative to the parent element

**REM**: Relative to the root element (HTML tag)

**%**: Relative to the parent element

**VW**: Relative to the viewport’s width

**VH**: Relative to the viewport’s height

Unlike PX, relative units like %, EM, and REM are better suited to responsive design and also help [meet accessibility standards](https://elementor.com/blog/wordpress-accessibility-elementor/). Relative units scale better on different devices because they can scale up and down according to another element’s size.

1em = 16px (1 \* 16)

2em = 32px (2 \* 16)

.5em = 8px (.5 \* 16)

1rem = 16px

2rem = 32px

.5rem = 8px

100% = 16px

200% = 32px

50% = 8px

### What is the difference between EM and REM?

Looking at the chart above, it shows EM and REM looking exactly the same. So how do they differ?

Simply put, they differ based on inheritance. As mentioned, REM is based on the root element (HTML). Every child element that uses REM will then use the HTML root size as its calculation point, regardless of whether or not a parent element has any different sizes specified.

EM on the other hand, is based on the font size of the parent element. If a parent element has a different size than the root element, the EM calculation will be based off of the parent element, and not the root element. This means that nested elements that use EM for sizing can sometimes end up being a size that you didn’t anticipate. On the other hand, it does give you more fine-grained control if you need it to specify the size for a particular area of a page.

### So what about %, VW, and VH? What are they all about?

While PX, EM, and REM are primarily used for font sizing, %, VW, and VH are mostly used for margins, padding, spacing, and widths/heights.

To reiterate, VH stands for “viewport height”, which is the viewable screen’s height. 100VH would represent 100% of the viewport’s height, or the full height of the screen. And of course, VW stands for “viewport width”, which is the viewable screen’s width. 100VW would represent 100% of the viewport’s width, or the full width of the screen. % reflects a percentage of the parent element’s size, regardless of the viewport’s size.

Let’s look at some examples of where Elementor gives %, VW, and VH options.

**Column Widths**: If you edit the layout of an Elementor Column, you’ll notice that there is only one width sizing unit available – %. Column widths only work well and responsively when using percentages, so no other option is given.

**Margins**: A section’s margins can be specified either in PX or %. Using % is usually preferable to ensure the margins don’t get larger than the content when scaling down for a mobile device for instance. By using a percentage of the width of the device, your margins will remain relative to the size of the content, which is almost always preferable.

**Padding**: A section’s padding can be specified either in PX, EM, or %. As with margins, it is often preferable to use either EM or % so the padding remains relative as the size of the page scales.

**Font Size**: If you edit the typography of an element, such as a Heading, you’ll see four choices: PX, EM, REM, and VH