

CSS

CSS UNITS

ABSOLUTE UNITS

The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

Absolute length units are not recommended for use on screen, because screen sizes vary so much.

PIXELS are the basic building block for everything in CSS. Strangely enough, **pixels** shouldn't be the unit of choice most of the time. **Pixels** are very straightforward, they say what they are going to do. And because of that, it is very tempting to use them. This will bite you if you continue to use pixels throughout your entire application. This is because pixels are really bad for accessibility.

PIXELS (px)

```
border: 2px solid red;
```



RELATIVE UNITS

Relative length units are relative to something else, perhaps the size of the parent element's font, or the size of the viewport. The benefit of using relative units is that with some careful planning you can make it so the size of text or other element scales relative to everything else on the page. These are the most useful units for web development.

PERCENTAGES %

A length set in % is based on the length of the same property of the parent element.

For example, if an element renders at 450px width, a child element with a width set to 50% will render at 225px.

% get a lot of use in responsive design and are pretty straightforward.

PERCENTAGES (%)

```
width: 50%;
```

EM

The **em** does not have a full form. It is the unit which is relative to the font-size of the current element so this may be different for each element of the HTML document.

For bigger screens, the font-size will look bigger while for smaller screens, the font-size will be relatively smaller.

The **em** unit is commonly used with padding, margin, text-indent etc.

EM

```
font-size: 1em;
```

REM

rem unit means "The root element's font-size". (**rem** stands for "**root em**".)

The `` elements inside the `` with a class of **rem**s take their sizing from the root element (`<html>`). This means that each successive level of nesting does not keep getting larger.

REM

```
font-size: 1rem;
```

VIEWPORT UNITS

The **vw** and **vh** units, which stand for viewport width and viewport height, are the last units that we will go over. No matter where you are, inside a child or in a grand grandchild, the **vw** and **vh** will always be the width and height of the viewport. The **vw** and **vh** don't care about the size of its parent.

VIEWPORT HEIGHT (vh)

This unit is based on the height of the viewport. A value of **1vh** is equal to 1% of the viewport height.

VH

```
height: 50vh;
```

VIEWPORT WIDTH (vw)

This unit is based on the width of the viewport. A value of **1vw** is equal to 1% of the viewport width.

VW

```
width: 30vw;
```

USE CASES FOR DIFFERENT UNITS

HOW TO PICK WHICH CSS UNIT TO USE AND IN WHICH CIRCUMSTANCE?

WELL, IT DEPENDS...

There are always exceptions, these are some ideas:

Are you declaring a font-size: you probably want to go with **rem** because they adapt to the user system and browser preferences.

Are you declaring a width: for the most part a **%** is a good way to go, sometimes is also good to use **vw** but be careful with **vw** because is the width of the browser window excluding the UI.

Are you declaring a height: first is actually to think if you really need a height, it's not always necessary and it would be good if you could use **min-height** (it helps the content expand without overflowing and causing issues on mobile) you could use **%, rem or vh.**

Are you declaring padding or margin: for the most part you should use either **em or rem.**

Are you declaring a border: is common practice to use **px.**

Are you declaring media queries: it would be good to use **em** for browser compatibility. *Safari sometimes does strange things with rem's in media queries.

ANY QUESTIONS?