

Lesson:

Introduction to the CSS Unit



Topics Covered

- What is a CSS unit?
- Absolute units
- Relative units

What is a CSS unit

CSS units are used to specify the size, length, and other measurements for various properties like font size, margin, padding, and more. There are several types of units available in CSS, and each has its own unique characteristics and uses. For example, if you wanted to set the property margin of a paragraph, you would give it a specific value. This value includes the CSS unit.

Look at this simple example:

```
p {
margin: 20px;
}
```

In this case, the **margin** is the **property**, **20px** is the **value**, and **px** (or “pixel”) is the **CSS unit**.

Even though it’s common to see units like **px** used, the big question is often, “What’s the best unit to use here?”

When considering all the options for which units to use, it’s important to consider the two categories of units: absolute and relative.

Absolute units

These are the **fixed-length units**, and the length expressed using the absolute units will appear as **exactly that size**. It is **not recommended to use on-screen**, because the size of the screen varies too much.

Absolute units can be useful when working on a project where **responsiveness is not being considered**. For **example, desktop apps** that can’t be resized can be styled to the default dimensions. If the window doesn’t scale, you don’t need the content either.

Pixels (px): It is used to define the measurement in pixels.

Example: “`font-size: 16px`” sets the font size to 16 pixels.

Centimeter (cm): It is used to define the measurement in centimeters.

Example: “`width: 5 cm;`” 1 cm equals 254 px.

Millimeters (mm): It is used to define the measurement in millimeters.

Example: “`width: 5cm;`” 1 cm equals 2.44px.

Inches (in): It is used to define the measurement in inches.

Example: “`height: 1in;`” 1in equals 245.51px.

Points (pt): A unit of length commonly used in typography to specify the size of text. One point is equal to 1/72nd of an inch, and the size of a point is typically defined as the height of the letter "M" in a given font.

Example: "font-size: 12pt;"

Picas (pc): It is commonly used in typography to specify the size of the text. One pica is equal to 12 points, or 1/6th of an inch.

Example: "font-size: 1pc".

Relative units:

Relative units are good for styling the responsive site because they scale relative to the window size or the parent (depending on the unit). They specify the length, which is relative to another length property.

Relative units can be a little more difficult to determine than absolute units, so let's go through your options in detail.

percentage (%): it is used to define the measurement as a percentage that is relative to the parent element's value.

Example: If the width of the parent element is 200 px, "width: 50%" sets the width of the child element to 100 pixels (50% of 200 pixels).

parent element's font size (em): It is relative to the font size of the parent element.

If the font size of the parent element is 16 pixels, then 1em would be equal to 16 pixels. If the font size of the parent element is 20 pixels, then 1em would be equal to 20 pixels.

Example: if the font size of the parent element is 16px, "height: 2em" sets the height to 32 pixels (2 times 16 pixels). And "width:4em" set the width to 64 pixels (4 times 16 pixels) of the child element.

index.html

```
<div class="parent">
    parent
    <div class="child">child</div>
</div>
```

style.css

```
.parent {
    font-size: 16px;
}

.child {
    height: 2em; // 2 * 16 = 32
    width: 4em; // 4 * 16 = 64
}
```

Browser output:

parent

div.child 64 × 32

child

The root element's font size (rem): it is relative to the font- the size of the root element.

Example: If the font size of the root element is 16px,

"font-size: 1.5rem" sets the font size to 24 pixels.

Index.html

```
<ul class="rems">
  <li>One</li>
  <li>Two</li>
  <li>Three
    <ul>
      <li>Three A</li>
      <li>Three B
        <ul>
          <li>Three B 2</li>
        </ul>
      </li>
    </ul>
  </li>
</ul>
```

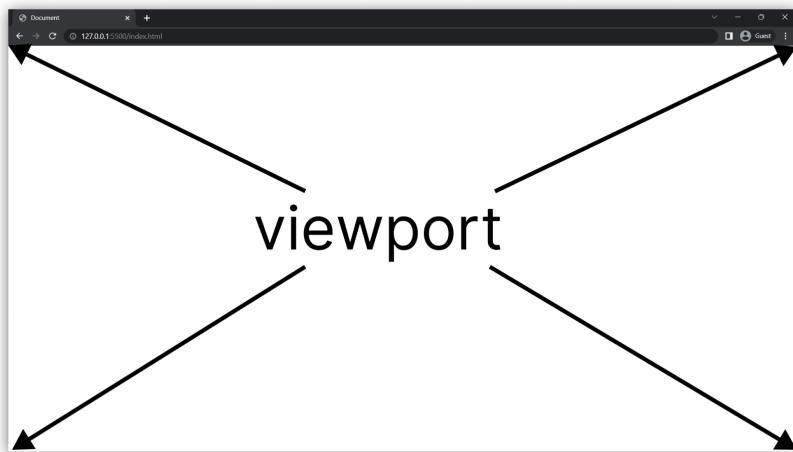
style.css

```
html {
  font-size: 16px;
}
.rems li {
  font-size: 1.5rem;
}
```

- One
- Two
- Three
 - Three A
 - Three B
 - Three B 2

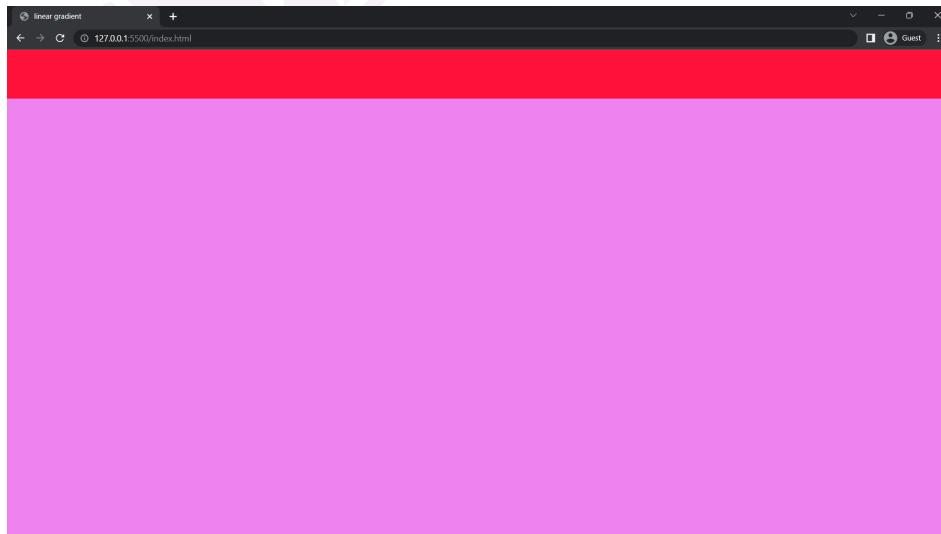
For understanding CSS viewport properties like **vh**, **vw**, **vmin**, and **vmax**, let's first understand what the viewport is.

"**Viewport**" is a term used in web development that refers to the visible area of a web page on a device's screen. It's basically the portion of the screen that you can see when you open a website on your phone, tablet, or computer.

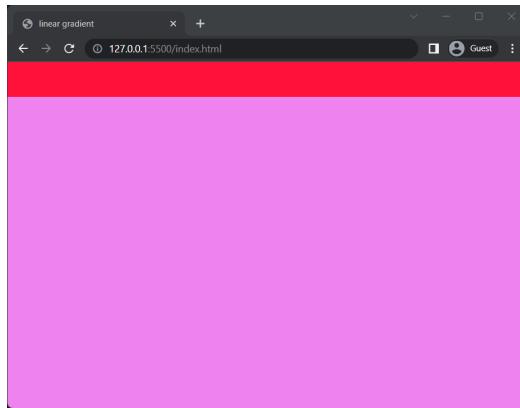


On a desktop device, the viewport size matches the **browser's window size, excluding the toolbar and other elements** that aren't part of the page. On a mobile device, the viewport is generally the size of the device's screen.

Let's say we have a webpage that has a header and a main section. The header is set to have a height of 10% of the viewport height, and the main section is set to have a height of 90% of the viewport height.



When we open the webpage in a maximized browser window, the viewport will be the full height and width of the scene. The header will take up 10% of the scene height, and the main section will take up the remaining 90%.



However, if we resize the browser window to be smaller, the viewport will also become smaller. As a result, the header and main section will also become smaller, since their heights are based on the viewport height.

CSS absolute units based on viewport

Viewport height (vh): It is relative to the height of the viewport.

1 vh = 1%, or 1/100 of the height of the viewport.

Example: “height: 50vh” sets the height of the element to 50% of the viewport height.

Index.html

```
<div></div>
```

style.css

```
div {
    background-color: violet;
    height: 50vh;
}
```

Browser output



Viewport width (vw): It is relative to the width of the viewport.

1 vw = 1%, or 1/100 of the width of the viewport.

Example: “width: 50vw” sets the width of the element to 50% of the viewport width.

Index.html

```
<div></div>
```

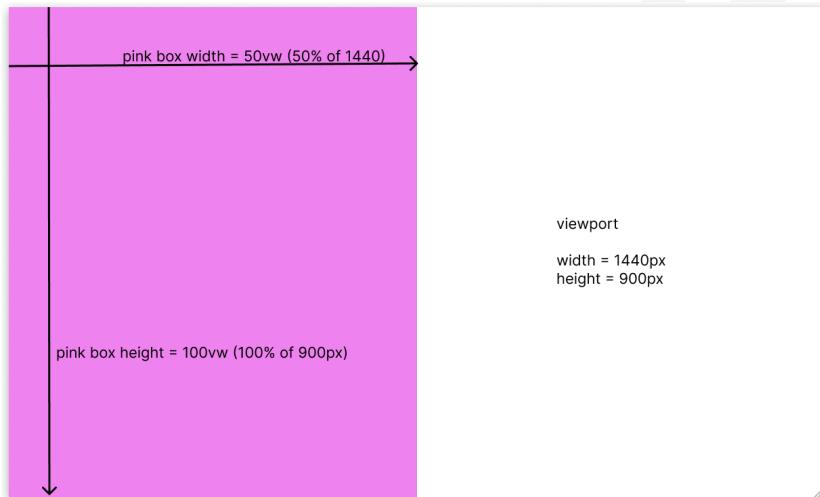
style.css

```
div {
    background-color: violet;
    width: 50vw;
    height: 100vh;
}
```

Browser output:

Viewport minimum (vmin): it is relative to the smaller dimension of the viewport (either the height or width, whichever is smaller) of the viewport.

Vmin example in mobile: If the viewport is 400 pixels wide and 800 pixels tall, the smaller dimension would be the width (400 pixels). Therefore, the width and height of the "div" element would be set to 400 pixels (100% of 400 pixels).



Viewport minimum (vmin): it is relative to the smaller dimension of the viewport (either the height or width, whichever is smaller) of the viewport.

Vmin example in mobile: If the viewport is 400 pixels wide and 800 pixels tall, the smaller dimension would be the width (400 pixels). Therefore, the width and height of the "div" element would be set to 400 pixels (100% of 400 pixels).

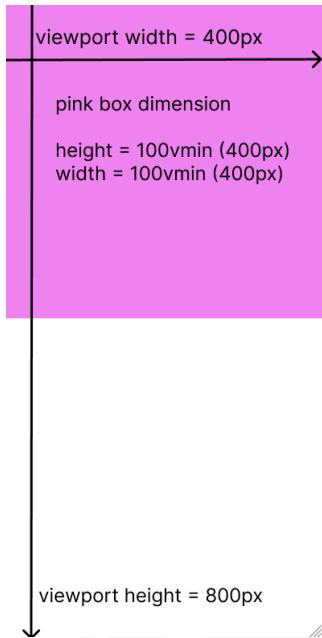
Index.html

```
<div><div>
```

style.css

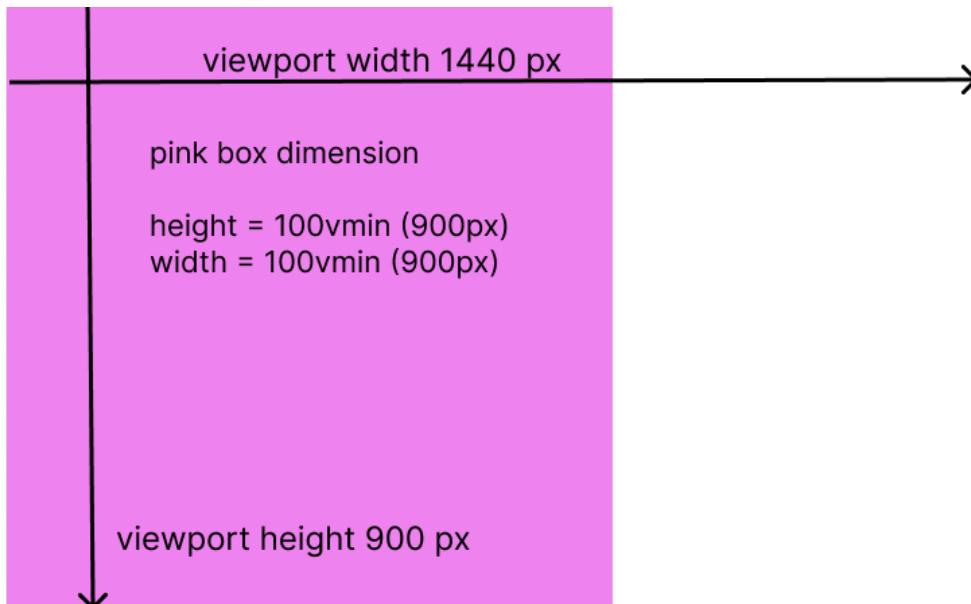
```
div {
    background-color: pink;
    height: 100vmin;
    width: 100vmin;
}
```

Browser output:



vmin example in desktop: If the viewport is 1440 pixels wide and 900 pixels tall, the smaller dimension would be the height (900 pixels). Therefore, the width and height of the "div" element would be set to 900 pixels (100% of 900 pixels).

Browser output:



Viewport maximum (vmax): It is **relative to the larger dimension of the viewport** (either the height or width, whichever is larger) of the viewport.

$1\text{vmax} = 1\%$ or $1/100$ of the viewport's larger dimension

Vmax example in mobile: If the viewport is **400 pixels wide** and **800 pixels tall**, the **larger dimension would be the height** (800 pixels). Therefore, the width and height of the "div" element would be set to 400 pixels (50% of 800 pixels).

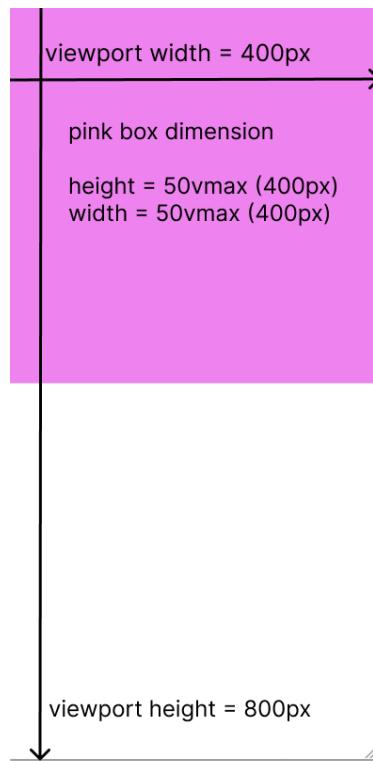
Index.html

```
<div><div>
```

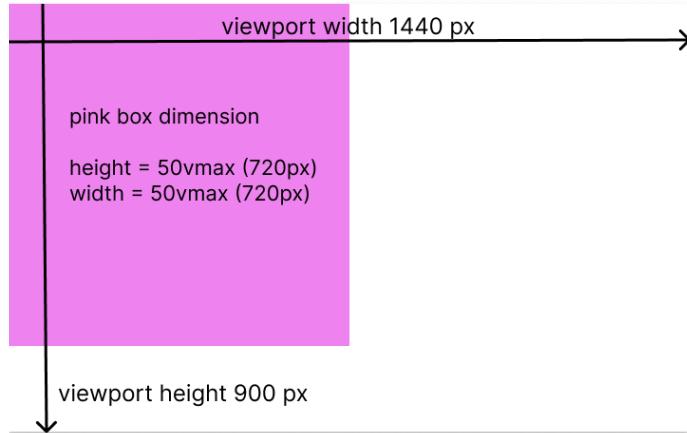
style.css

```
div {  
    background-color: pink;  
    height: 50vmax;  
    width: 50vmax;  
}
```

Browser output:



vmax example in desktop: If the viewport is **1440 pixels wide** and **900 pixels tall**, the **larger dimension would be the width** (1440 pixels). Therefore, the width and height of the "div" element would be set to 720 pixels (50% of 1440 pixels).

Browser output:

character (ch): This unit is based on the width of the "0" character in the font used for the element.

Example: "width: 10ch" sets the width of the element to the width of 10 "0" characters.

X-height (ex): It is relative to the x-height of the font of the element. It is rarely used, the x-height is determined by the height of the lowercase letter "x".

Example: "font-size: 2ex" sets the font size to twice the x-height of the font.