What are vh and vw units, and when should you use them?

In CSS, vh and vw are viewport-relative units that allow you to size elements based on the dimensions of the browser window. These units are particularly useful for creating responsive designs that adapt to different screen sizes.

vh stands for "viewport height," and 1vh is equal to 1% of the viewport's height.

Similarly, vw stands for "viewport width," and 1vw is equal to 1% of the viewport's width.

This means that if you set an element's height to 100vh, it will occupy the full height of the viewport, regardless of the actual pixel dimensions of the device.

These units are especially handy when you want to create full-screen layouts or elements that maintain a specific proportion of the screen.

For example, you might want to use them to create a hero section that always fills the entire screen:

.hero {

height: 100vh;

width: 100vw;

}

This CSS ensures that the hero section will always be exactly the size of the viewport, regardless of the device's screen size.

vh and vw units can also be used for typography to create responsive text sizes. For instance:

h1 {

font-size: 5vw;

}

This will set the font size of h1 elements to 5% of the viewport width, allowing the text to scale smoothly with the browser window size.

One of the advantages of vh and vw units is that they respond to changes in the viewport size in real-time. This means that if a user resizes their browser window, elements sized with these units will adjust accordingly without needing to reload the page. However, it's important to use these units judiciously. Setting font sizes solely with vw units, for example, can lead to text becoming too small on narrow screens or too large on wide screens.

Another consideration is that on mobile devices, the viewport height can change when the browser's address bar appears or disappears, which can cause unexpected layout shifts if you're using vh units extensively.

In summary, vh and vw units are powerful tools for creating responsive layouts and elements that adapt to the viewport size. They're particularly useful for full-screen sections, maintaining aspect ratios, and creating smoothly scaling designs. However, they should be used thoughtfully and often in combination with other CSS techniques to ensure the best user experience across all devices.