Instead of using **em or px to size text**, you can use **viewport units** for responsive typography. Viewport units, like percentages, are relative units, but they are based off different items. Viewport units are relative to the viewport dimensions (width or height) of a device, and percentages are relative to the size of the parent container element.

The four different viewport units are:

* **vw (viewport width):** 10vw would be 10% of the viewport’s width.
* **vh (viewport height):** 3vh would be 3% of the viewport’s height.
* **vmin (viewport minimum):** 70vmin would be 70% of the viewport’s smaller dimension (height or width).
* **vmax (viewport maximum):** 100vmax would be 100% of the viewport’s bigger dimension (height or width).

Here is an example that sets a body tag to 30% of the viewport’s width.

body {

font-size: 30vw;

}

A large image can be perfect on a big computer screen, but useless on a small device. Why load a large image when you have to scale it down anyway? To reduce the load, or for any other reasons, you can use media queries to display different images on different devices.

For example:

/\* For large screens: \*/

body {

background-image: url('https://via.placeholder.com/1200');

}

/\* For width less than 720px: \*/

@media only screen and (max-width: 720px) {

body {

background-image: url('https://via.placeholder.com/500');

}

}

Notice, how in the above code two background images are used of different sizes and are applied as the background of the whole page. Only once get loaded at one time, based on the size of the screen.

You can use the media query max-device-width, instead of max-width, which checks the device width, instead of the browser width. Then the image will not change when you resize the browser window, instead it will only function for the devices.