* Use CSS grid and/or CSS flex layout: By default CSS grid and CSS flex layout are responsive. CSS grid will adjust the width of the layout to fix the width of the device. While CSS flex has a flex-wrap property that would wrap a container and move it below the page if the page width is smaller than the width of the container. The point of responsive design is displaying content on any device properly. Content on a responsive website would look good on any screen size. CSS grid and CSS flex properties gives us this freedom with little or no media queries. CSS grid and CSS flex properties are very easy and pretty straight forward to learn.
* Build for mobile first: Mobile first approach is recommended because it is easy to adjust a website UI from a small view to a larger view. Besides, the @media screen and (max-width :;) property, use for building for desktop to mobile, can be overwritten easily. For example, if you place @media screen and (max-width: 300px ;) above @media screen and (max-width: 500px ;) the code in the latter will overwrite the formal, and this would be disastrous. But this is not the case for the @media screen and (min-width :;) use for building for mobile to desktop.
* Use min-width 0r max-width for container instead of fixed values (height: 300rem): Min-width is the minimum width of a container while the max-width is maximum width of a container. Using these CSS properties will decrease the number of media query one would write. For Example:

.container {

Max-width: 500rem;

}

Would not grow larger than 500rem in width on a larger device.

.container {

Min-width: 10rem;

}

Will not go small than 10rem in width on a smaller device. The above is preferable when building a responsive apps.

-Use % unit, vw, rem unit: Using percentage and vw unit is the preferable unit of width value. A container that is set to width 50% or 50vw will occupy 50% or 50vw of the page on all screen size. It is a relieve cause we would not need to write different sizes of a container for different screen size. To make it better, use % unit with max-width. Setting a container to a width of 100% will occupy a page 100% as we’ve discuss on all screen size, but then adding max-width of 30rem means the container would not get wider than 30rem on a large screen but on a small screen it would occupy the width 100%. With this approach we would write less media queries. The VW means view port width. The entire page width is the view port width to put simply. Setting our body element, container, text or image to a view port width means the element will not overflow the page or screen regardless of the screen size. Which comes in handy.

Rem unit are relative unit. 1rem = 16px. Use rem unit when writing fixed value for padding, margin or height.

-Set the body overflow-x property to hidden: setting the body overflow-x property to hidden would reduce the anxiety of scrolling horizontally especially when one have a very large image. Setting this property hidden would hide all the content that have overflowed the width of the body or page. Unless we want to write code for horizontal scroll effect, then we can set the overflow-x property to auto or scroll.