media queries 20MIN

Breakpoint

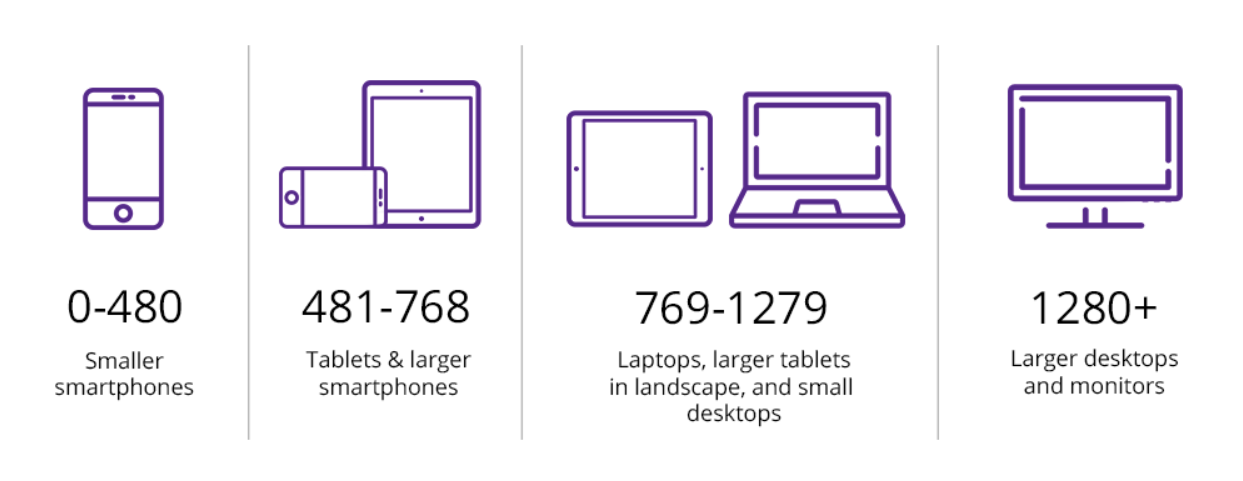
Introduction of breakpoints

You: Well, I've come across the term "breakpoint" in web development, particularly in relation to responsive design. I wanted to know more about it.

Newton: Ah, breakpoints! They play a crucial role in creating responsive websites. In simple terms, breakpoints are specific points in a website's layout where the design adapts and responds to different screen sizes or devices. It allows the content to be displayed optimally regardless of whether it's viewed on a desktop, tablet, or smartphone.

Some Common breakpoints for devices

Because they are used with media queries, CSS breakpoints are sometimes known as media query breakpoints.



<div class="container">

<h1>Welcome to my website!</h1>

<p>This is an example of a responsive design with breakpoints.</p>

</div>

.container {

width: 100%;

max-width: 960px;

margin: 0 auto;

}

@media (min-width: 768px) {

.container {

width: 80%;

}

}

@media (min-width: 1200px) {

.container {

width: 60%;

}

}

In this example, we have a container with a maximum width of 960 pixels. However, using media queries, we set different widths for the container at specific breakpoints. When the screen width is at least 768 pixels, the container's width becomes 80%. Similarly, when the screen width reaches 1200 pixels or more, the container's width changes to 60%. This allows the website's layout to adapt to different screen sizes.

Viewport

The viewport defines the area within a browser window where web content is displayed. It influences the initial layout and the way users interact with a website.

Viewport

👉A web page's viewport is the portion that the user can see.

👉On a mobile phone as opposed to a computer screen, the viewport is smaller because it fluctuates depending on the device.

👉Before tablets and smartphones, online pages were solely intended to be seen on computer screens, and they frequently had a static design and a set size.

Real Life Example

Imagine you're visiting a website on your smartphone. The screen size of your phone is much smaller than that of a desktop computer. Without a properly defined viewport, the webpage might appear zoomed out or require constant scrolling, making it difficult to navigate and read the content.

Controlling the viewport via meta tag:-

HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag.

<meta name="viewport" content="width=device-width, initial-scale=1.0">

It gives the browser instructions on how to control the page's dimensions and scaling. The width of the viewport is not always the width of the window.

Properties

CSS provides several properties and units that enable responsive design and adaptability to different viewports. Here are some commonly used CSS properties related to the viewport:

👉Width and Height: The width and height properties allow you to set the dimensions of elements relative to the viewport. For example:

div{

width:"100vw";

height:"50vw";

}

👉Min-Width and Max-Width: These properties set the minimum and maximum width that an element can be, ensuring it remains within a specific range in relation to the viewport. For example:

div{

min-width:"100px";

max-height:"50px";

}

👉Min-Height and Max-Height: Similar to min-width and max-width, these properties define the minimum and maximum height for an element relative to the viewport.

👉Media Queries: Media queries allow you to apply different CSS styles based on the characteristics of the viewport. By specifying certain conditions, you can target specific screen sizes or device orientations. For example:

@media (max-width: 600px) {

55 styles for screens smaller than 600px

}

@media (min-width: 601px) {

CSS styles for screens larger than 600px

}

👉Viewport Units: CSS provides viewport-relative units that allow you to size elements based on a percentage of the viewport dimensions. The two commonly used units are viewport width (vw) and viewport height (vh). For example:

div{

min-width:"50vw"; 50% of the viewport width.

max-height:"50vw"; 50% of the viewport height.

}.

Orientation

Real Life Example

Imagine you're designing a website for a clothing store. You want to display different categories of clothing, such as shirts, pants, and accessories. By using orientation, you can arrange these categories in a way that makes it easy for visitors to navigate. For example, you might choose a vertical orientation with the categories listed in a sidebar on the left side of the page.

Values:-

Orientation feature can be defined by specific values which are given below -

portrait - The viewport is in a portrait orientation, i.e., the height is greater than or equal to the width.

landscape - The viewport is in a landscape orientation, i.e., the width is greater than the height.

Code Based Example

<div>Box 1</div>

<div>Box 2</div>

<div>Box 3</div>

body {

display: flex;

}

div {

background: blue;

width: 200px;

height: 100px;

margin: 0.5rem;

padding: 0.5rem;

}

If viewing on a mobile device, the boxes will be in column flex and when viewing in landscape mode, the boxes will be in row order.

@media (orientation: landscape)

{

body {

flex-direction: row;

}

}

@media (orientation: portrait) {

body {

flex-direction: column;

}

}