Sizing Structural Elements Responsively

Daniel Quinn

So, what we'd like to do in this tutorial is demonstrate a little bit of responsive design in HTML, a very basic approach in terms of putting together a few div elements that will resize as the screen resizes. And we're going to use a floating method to do that, which is an older way of doing things nowadays. But for the purposes of understanding the principles of responsive design, we'll use that because it's simpler. In the production environment, you'll probably want to work with Flexbox instead of a float method. So, let's just get started and try that out.

The first thing we'll want to do is set up our markup so, you know, we have something to work with. So, we're just gonna create a basic structure of elements to work with. So, make a bounding element that binds the whole thing. And then, we're going to put an inner element that's meant to contain the maximum width of the screen. And we're going to put two elements inside of that. One will be our content area, and one will be a sidebar. And we want these to sit next to each other. So, let's just call that content. And then we're going to call the next one sidebar.

And just for the sake of being able to see what's inside of this stuff, we'll get some lorem ipsum. So, I like to go to a lorem ipsum website, which will generate some random text. And we'll just grab a paragraph or two. And we're gonna drop it into our content div. And in the sidebar, let's put maybe a heading that says this is where widgets are going to be and maybe a couple of--a list element that will have some items in it. We'll just put a few to make things long. And we'll copy the same widget. So, we probably want to wrap our widget inside of a container, just so that if we want to target it later, we could. OK. Let's scoot this guy over and tap him. OK. Let's take a look at what we have.

So, now I've got these two divs; they're just sitting one underneath the other. So, what we'll need to do now is attach a style sheet so we can actually start writing up the styles. So, what we'll do is make a new file. We'll call it Style that CSS. We'll go back to our HTML, and we're going to attach that style sheet. So, what we'll do here is, in the header, we'll link to it. So, we're going to call it Style that CSS. And now it's attached. When we switch over to the styles, what we'll want to do is set up the bounding elements so that it wraps the two.

There are a number of ways to make sure that the floats don't interfere. An overflow is one way to do that. And then, let's actually set up the two elements so that they wrap. So, first, what we want to do is give our inner container a maximum width because we don't want it to fit to the entire screen. So, we're going to say this

maximum width is 1024. And what we're going to do is make sure that that gets centered in the screen. And we'll give it a background color, just so they can see it.

Then, what we want to do is determine the width of the content div and the sidebar div. So, what we'll do here is set content. And we're going--we want the div to actually be like 724 pixels. But when we want this to be responsive, we're gonna have to turn this into a percentage. And we'll do that in just a second. We know we want it to float to the left, and we want to give it a background color, so we can see it separate from everything else.

Now, we need--so we now have them floating next to each other. We need to define the size of the sidebar. So, what we'll do is say bounds, bounds wrap, and then sidebar. And that width is about 300, we're going to say. And we're also going to float it to the left, and we're gonna give it a background color that will let us identify it. So, you'll see here that the two elements are falling under each other. So, that's only because our panel here isn't big enough in the screen. So, if we look at it, we'll see if they're still floating.

Now, what we want to have happen is when we scale this down, we want the two elements to scale down as well instead of just fall under each other. And then eventually, when we get to a certain breakpoint, like, say, about 600, we want them to sit under each other and be 100 percent. So, what we need to do is, instead of defining these as pixels, we need to define them as percentages. And we want to make sure that the percentage we're basing it off of still matches that context ratio, which is this 1024 pixels.

So, an easy way to do that is to just do a quick division. So, we want to say that our 724 pixels is divided by our context ratio, and then we're going to times that by 100. And you're going to get a weird percentage out of that, which is OK. So, what we're going to end up with is this about 70%. And this will keep it pretty much pixel-perfect to what it was when it was pixels. And then we're gonna do the same thing for our sidebar. So, 1024 times 100. Come out to about 29%.

Let's take a look at what that's done. So, they're the same width and the same context ratio. But as we start to scale them, they start shrinking as the screen shrinks too. Now, we need this to not happen forever. So, we need, at some point, when we get to about 600 pixels, and you'll see in Chrome here, it's showing you scale down, we want it to the widgets sidebar to fall underneath the content. So, that's where breakpoints come into play.