Media Queries CSS Syntax

Hello and welcome to another lesson about CSS media queries. In this video, I would like to focus on the exact syntax and explain how you can work with different screen sizes when writing your CSS code. Let's look at the syntax and the first part is @ role.

@Media Rule

So, @media and there are a few different @ rules in CSS and they basically provide CSS with instructions on how to behave and for the @media role, the next part is - media type. It defines what type of media we are trying to target and we have a few values and types of media we can target including: all - that would match all the devices. Screen - matches devices with a screen, probably the most common thing you'll see. Or print - matches documents that are viewed in a print preview.

Older Browser Support

You might also see the word 'only' as a @media - only screen. For example, only here is used to prevent older browsers that do not support media queries with media features from applying the specified styles. I would say it is no longer needed and usually I just leave this out, as well as I left out the entire media type. When it is not specified, it will default to all, okay? But just let's look at using all for now.

Media Feature - Width

And next, we define media features. In this video, I will only focus on a screen width, but there are many, many more features we could target.

And media features can be connected with logical operators and the most common being 'and' so when we define media type, we also need to use the logical operator 'and' here to connect it with some media feature. I will refer you to an excellent article on CSS tricks if you're interested in digging deeper. But for this lesson, let's start with the basics, which is also everything you should need at the beginning. Now that we have our @ role, our media type, and the logical operator, we can specify the media feature that applies to screen width and that can be min-width or max-width. We have to use it inside round brackets. So min-width: and then our value for the size it can be in pixels, or a different unit. So let's try with 400 pixels. So that's our media feature. We could do another logical operator 'and' for example, and specify some other feature. But let's keep it simple for now. So we have media all and min-width 400 pixels and next we are opening and closing curly brackets. So, what that means is - now the browser, when it reads our CSS, it will find that @ media rule and it has to figure out, does it meet the condition? So, if the place that the web page is being displayed, does it use that media type? Well, it does because it applies to everything. And does the minimum width is at least 400 pixels, if it is, then the browser reads the CSS that is inside here, if it doesn't meet that condition, so if the screen width is less than 400 pixels, it will disregard everything that's inside here. Okay, so let's try it with H1 color, red, okay? And now you see this screen is a minimum width of 400 pixels. But if I change it to smaller screen size, you see, now it's less than 400 and now this media query applies. Okay, so let's have a look at media queries used by the Divi theme. Okay, so this is the code I copied from one of the articles on Elegant Themes, we have some comments here that will describe the media queries for us. So, this one targets all devices with a minimum width of 1405 pixels and it says responsive styles, large desktop, and above. Next, we have styles for standard desktop only and for Divi that would mean it's between 1100 pixels and 1405 pixels. Okay, that's why we have this and min and max, okay, double condition. Here, tablets and below. So the maximum width is 980 pixels. Tablets only, minimum width 768 and maximum width 980. Smartphone only maximum width 767. So anything between 0 to 767 that would be the smartphone only styles, or if we go even smaller, that would mean, smartphone in portrait view. So, maximum width 479 pixels, these are the standard media queries used by the Divi theme. But the important thing here is also the order of our CSS declarations, the media query itself doesn't add any specificity value. So we need to remember that the normal CSS applies to every device. So by normal, I mean this which is not inside a media query that applies to all screen sizes, if we specify the normal declaration and then we want to change it for smaller screen sizes, for example, the media query, let's say this one - smartphone portrait. I want this to be blue. So that media query needs to be further down the page after the original selector. Because if we put the normal code, let's see on the smaller screen sizes. Okay, that works. But if I put "normal" standard selector without any media queries afterwards here, this doesn't overwrite, it's because it's in a media query. Like I said, there is no additional specificity just because it's inside a media query. It's not making the styles more important in any way. So the order of things, if we are using the same selector does matter. In other words, the browser reads our code and it goes like this. Okay, I am at maximum width 479 because my screen width is 300, for example. So I will make this H1 blue and then it will read on and now it is - Okay, H1 is actually this color so I will use this one instead. And another very important thing when working with media queries is to mind the brackets, media query wraps the CSS declarations in an additional set of curly brackets, okay? So this and this, very important. We do need to remember to close the opening bracket. Once we start adding our CSS rulesets. So this could look like this. Okay. And now we have two closing brackets next to each other - this one closes the media query. But where it can get confusing is when we are copying things from the Chrome inspector, let me show you.

Chrome Inspector

When we preview our CSS here in the styles box, in the Chrome inspector, it will tell us if the CSS ruleset is inside a media query, but as you can see here it doesn't display the curly brackets here. So if I copy that, okay, and when I paste it here, this isn't a correct media query. It needs the opening and closing brackets. Here is a simplified version but if we actually look at that code here. So if I open that file, and use the style formatter. That will show me that this is the CSS and it's inside that media query, okay? And it does have that opening and closing bracket at the end. You have to be careful when copying code from here with the media query. You have to remember to add that additional set of opening and closing curly brackets. It is very important and your code will not work otherwise.

Where to add CSS media queries?

So you can use media queries inside the theme option CSS in the page-specific CSS in the page settings, or within any CSS file, you are queuing, your child theme stylesheet or a custom CSS file loaded by the functionality plugin. You won't be able to add media queries inside the modules advanced tab, just like you cannot specify your own selectors here. So using media queries is the only way we can target specific screen sizes with CSS.

So hopefully, this overview will help you start using it when writing your code.