

# Common Mobile Issues and How to Solve Them

Hello and welcome to the final lesson in the Divi Mobile Responsiveness Module where I would like to share a few common problems you may encounter when trying to optimize your Divi websites for smaller screen sizes with a solution on how you can fix them.

## Navigation Wrapping

Here, we have a very common situation where the default navigation looks great on desktop but as soon as we resize the window and before it displays the mobile view, the navigation wraps to the second line, your menu items are displayed on two lines instead of one. And the most common solution you would see is to change the breakpoint between the default navigation and the mobile version sooner. But what I would also like to do is decrease the space between menu items and decrease the font size before I switch to a hamburger view. That gives me a few more pixels of the default navigation more before we display that mobile version. So, let me show you how I would approach this. First, we have to define, where is that break point for the top navigation here. As I resize, the browser window we can see, okay, that at 1097 pixels, 96 actually it breaks to a new line so that would define my breakpoint and now I can decrease the padding here, you see the lis item, each of the lis items has a right padding of 22 pixels. So let's use the same selector top-menu, li, padding-right: 15 pixels. Now, let's wrap this into our media query, so @media, max-width of 1096 pixels. And I also want to specify the min width, so I do not target anything when the navigation is in the

default mobile style, so min width is 981 pixels and that's the logical operator we need, Okay? So let's see how many more pixels that gives us. So I can still - now at the 1037 I could also decrease the font size. So let's try adding another media query. At 1037, let's change the font size of the <a> element. I think the default is 14, so let's try with 13 pixels, and that should give us a bit more. Here - 988, you see, now I only need to switch the navigation from the default to mobile version on 988 pixel width, which basically gave me almost a hundred pixels of the default navigation. So now, between 981 and 988, we want to make sure that this menu is not visible. So - display: none, and we want to display the mobile navigation instead. And we can find that selector actually right here, below the top menu, which is inside that nav, ID top\_menu\_nav, and below we have ID with the mobile\_nav\_menu. If I change the screen sizes, we can see that this is display block where the screen is at 980 pixels. So we want to show this sooner, okay? So back to my - back here. Display: block, okay, so now if I check that page, on desktop, it looks good, and then it doesn't wrap to a second line, and then my navigation shows almost at the same point as it would normally show. Now, let's look at that menu module. It will use a similar, but a bit different selectors. So first, again, let's define the breakpoint where that goes to the second line. So here is fine, and that is, way sooner because we have bigger font size here. So that would be okay, so at 1356 pixels our navigation breaks to a new line. So, let's target, let's see the spacing here. That works a bit differently, each of the li has right and left padding. et\_pb\_menu et\_menu li. We could replace the et\_pb\_menu with our own custom CSS class, but let's just target that generic menu module class. So for that menu module, we would define a different media query, so media, similarly min-width 981 pixel. We don't want to target anything that is already in mobile view, but the max-width was

1354, that was 56. And now, let's change the padding - padding 0 and then maybe 5 pixels on each side. This is a hyphen. Okay, so that brings that items maybe too close to each other, let's try with 8. So let's change the font size right away here at the same breakpoint so right now it uses 16 pixels. So let's change this, but I think we need a more specific selector. So, let's see. What is defining that font size here? That's our theme builder generated CSS. So back here. I do not recommend using that 0 here. We would be better replacing it, with a custom CSS class added to the module settings. But just to show you a real quick font size, 14 pixels. Okay, that's a good start. Now, let's check that breakpoint now. So from 1350, we went down to 1195. And that would be the breakpoint at 1189 or 90, we could switch to the mobile version. But that's still better than switching to mobile at 1300 pixels, right? So, similarly to this, that was 1190. And let's see, what are the selectors for the mobile menu and the default navigation, let's see here. We have two divs, `et_pb_menu`, `menu` and `et_mobile_nav_menu`. So, this one is flex on larger screen size for desktop, and this one is flex on mobile. Okay? So let's try that. Let's use that selector `display: flex` and let's use this and `display: none`. Okay, let's see. Okay, that's a good start. The alignment is incorrect. Let's see what other CSS is being applied to that mobile menu. `Justify-content: flex-end`. I think that is, what is moving that yes to the right side. So if we add that same selector just we want this to work sooner. `Justify-content: flex-end`. Okay, great. I think there are some alignment issues here. Let's see, what else are we missing? `Align-item: center`, that would be it. Okay, So for the same selector, `et_mobile_nav_menu`, `align-items: center` `et_mobile_nav_menu` `align-item: center`, perfect. Now it doesn't wrap to a second line and it displays correctly on every screen size.

## Text Wrap

Now, a different popular problem on mobile is when you're using a large font sizes. Your words can wrap and bleed onto a new line when you don't really want it. So obviously, the first advice would be to decrease your font size but if for any reason you can't do that or don't want to do that and there's just you know a little space missing to fit that text here, you can use a CSS property called overflow wrap. So here on that text module, I can, for example, target that div which is inside the main module, the et\_pb text inner. And now I can define overflow - wrap to normal which basically prevents the word wrap, but you have to be careful because that text will bleed outside of the container like this. You see? So on very small screen sizes, it can cause the horizontal scrolling, so it may not be the best thing to use, always, but just note that you have that option. A different thing you can try to do instead of the overflow wrap, you can hyphenate the text. So hyphens: auto, and that would add that hyphen and at least it will make sense. So either of the two can be helpful but probably decreasing the font size would be the best advice.

## Overlapping Elements

Since we are on the horizontal scrolling issue, this problem can appear on desktop too but it's more common on mobile that sometimes some elements will overlap the standard width of the page and on smaller screen sizes that can cause this horizontal scroll to be visible. Now, there is no simple solution to this but I want to show you how I would approach, if I wanted to inspect that page and see what is causing that horizontal scrolling, I would click that select element and I would kind of scroll through here and try to hover each of the elements because it can be a parent or a child element and just see which of this will - like this, you see. I can see that this element is outside of that width. And now, if I can look in the inspector, when I select that, I can see that it uses a

negative margin of 165 pixels. So to keep that a margin and keep that placement of that image, but remove that horizontal scrolling, we need to add an overflow property on the section. So it is inside the Divi Builder, Advanced tab, overflow, horizontal overflow, we can set it to hidden. So overflow-x: hidden, and that, no longer scrolls on the side. Overflow hidden basically means that any inner content that's bigger than the actual element will not be visible. Only the actual size of that element is shown with the overflow:hidden.

I hope you'll find this advice helpful and you'll see how powerful the browser inspector is using media queries and CSS. Don't be afraid to look into your page source. Many times you will need to target the different elements to make sure they have the correct size and alignment on a certain screen width. Being able to define custom media queries and correct selectors is a very important skill we need to practice.