Responsive Breakpoints

Since Bootstrap is developed to be mobile first, we use a handful of media queries to create sensible breakpoints for our layouts and interfaces. These breakpoints are mostly based on minimum viewport widths and allow us to scale up elements as the viewport changes.

Bootstrap primarily uses the following media query ranges – or breakpoints – in our Sass files for our layout, grid system, and components.

// Extra small devices (portrait phones, less than 576px)

// No media query for 'xs' since this is the default in Bootstrap

// Small devices (landscape phones, 576px and up)

@media (min-width: 576px) { … }

// Medium devices (tablets, 768px and up)

@media (min-width: 768px) { … }

// Large devices (desktops, 992px and up)

@media (min-width: 992px) { … }

// Extra large devices (large desktops, 1200px and up)

@media (min-width: 1200px) { … }

These five different sizes and names come up again and again and again in Bootstrap, and when we will talk about the Grid System, we will use them all the time. We will say make these two columns extra small and five columns on large. But now we will see how we can use these on the spacing utilities of Bootstrap, so we selectively add margin and padding depending on the screen size. The users normally want their items to be more spaced out on a large screen and no space on a small screen. We can do that.

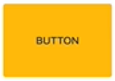
We had been using the following format {property}{sides}-{size} for xs

For example, ml-5 (marginleft-5) for xs

And there is also {property}{sides}-{breakpoint}-{size} for sm, md, lg, and xl.

So, we can say once we hit medium size, we want padding to have 5px on the top or once we are at large we want the margin to be zero at all four sides.

```
<button class= “btn btn-warning p-5”>BUTTON</button> // We are starting with padding 5px at all sides
```



Now we only want this button to have 5px padding not on all sides but at only extra large. Then we will write our statement as below.

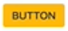
```
<button class= “btn btn-warning p-xl-5”>BUTTON</button>
```

After this statement executes, the button will only have padding 5px at all sides once the screen hits extra large, if it goes below extra large then the button will return to its original position.

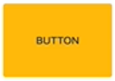
Now if we change the button to have 5px padding at all sides once the screen is at small size then we write our statement as below.

```
<button class= “btn btn-warning p-sm-5”>BUTTON</button>
```

Now when we hit extra small (xs) the button will not have padding of 5px at all sides as below



But then when we hit small, we will have padding of 5px at all sides as shown below

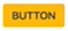


And it will stay that way even if we go up from small size, so its not saying that it will only show padding of 5px once we hit small size. Its on all sizes small and above.

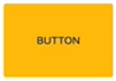
Now if we want the button to have padding at sizes larger than small then we must specify that in our statement as below.

<button class= “btn btn-warning **p-sm-5 pm-md-0**”>BUTTON</button>

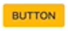
Now when we are at extra small the button has no padding as below



The when we hit small it has padding of 5px at all sides as below



Then from there when we hit medium the padding again goes away and displays as below



So, we can keep on specifying padding for different sizes and the site will show the padding according to the screen they are on.

We can have an alternate like

<button class= “btn btn-warning **p-sm-5 p-md-0 p-lg-5 p-xl-0**>BUTTON</button>

It means having padding of 5px at all sides once we are at small size, then have no padding when we hit medium, then again have padding when we hit large, and then again not having padding when we hit extra large.

One more thing we must note that there is no notation for extra small. So, if we want to define a padding for extra small sizes of screens then we write like below

<button class= “btn btn-danger **p-1 p-sm-2 p-md-3 p-lg-4 p-xl-5**>BUTTON</button>

So, as you can see, we are having padding of 1px at all sides, but we are not defining the size with any notation like p-xs-1, instead we are just writing p-1.

The classes are named using the syntax {property}{sides}-{size} for xs and {property}{sides}-{breakpoint}-{size} for sm, md, lg and xl.

So, whatever we apply if we do not specify a breakpoint it defaults the padding to everything including extra small to extra large, that's why it's the developer's job to correctly specify at what size how much padding he/she wants.

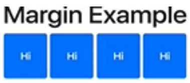
```
<button class= "btn btn-success p-0 pl-sm-5 pt-md-5 pr-lg-5 pb-xl-5">BUTTON</button>
```

According to the statement above, we have no padding on our button when the screen is extra small, we have 5px padding on the left from the center of the button when the button is small, we have 5px padding on the top from the center of the button when the screen is medium, we have 5px padding from the center of the button to right when the screen is large, we have 5px padding from the center of the button to the bottom when the screen is extra large.

The same rule applies with margin,

```
<h1>Margin Example</h1>
<button class= "btn btn-primary p-4">Hi</button>  //Having margin of 4px on all four sides.
<button class= "btn btn-primary p-4">Hi</button>
<button class= "btn btn-primary p-4">Hi</button>
<button class= "btn btn-primary p-4">Hi</button>
```

Output:



Margin Example

Hi Hi Hi Hi

Ok so lets just start working with left and right margin

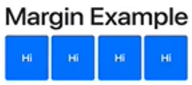
```
<h1>Margin Example</h1>
<button class= "btn btn-primary p-4 mx-xl-5">Hi</button>
<button class= "btn btn-primary p-4 mx-xl-5">Hi</button>
<button class= "btn btn-primary p-4 mx-xl-5">Hi</button>
<button class= "btn btn-primary p-4 mx-xl-5">Hi</button>
```

The above statements mean that once the screen hits large, the blocks will have margin of 5px on both sides (left and right).

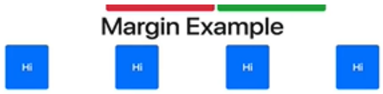
Here “m” represents medium, “x” represents left and right and “xl” represents extra large.

Output

At extra small, small, medium and large screens



At extra large screen

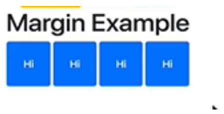


```
<h1>Margin Example</h1>

<button class= “btn btn-primary p-4 mx-0 mx-sm-2 mx-md-3 mx-lg-4 mx-xl-5”>Hi</button>

<button class= “btn btn-primary p-4 mx-0 mx-sm-2 mx-md-3 mx-lg-4 mx-xl-5”>Hi</button>

<button class= “btn btn-primary p-4 mx-0 mx-sm-2 mx-md-3 mx-lg-4 mx-xl-5”>Hi</button>

<button class= “btn btn-primary p-4 mx-0 mx-sm-2 mx-md-3 mx-lg-4 mx-xl-5”>Hi</button>
```

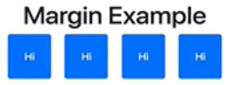
This means when we are at extra small screen there is no margin, then when we hit small the blocks will have margin of 2px, on medium they will have margin of 3px, on large they will have margin of 4px and finally on extra large they will have margin of 5px.

Output:

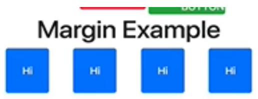
Extra small


Margin Example

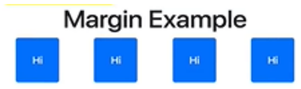
Small


Margin Example

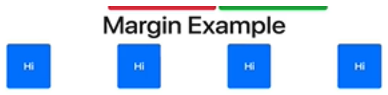
Medium


Margin Example

Large


Margin Example

Extra large


Margin Example