

Bootstrap grid system: <http://getbootstrap.com/css/#grid>

So, the next step is really to try and understand a bit of, how we can gain control over the layout of the content of our site. And Bootstrap has a sort of brilliant system for doing this, and this is where we really start to engage with the concept of responsive. So remember, we've already seen responsive in the way the navigation bar worked. Do you remember how when we shrunk down the page, the navigation bar disappeared, and showed us a special button to show and hide it. Well, we're now gonna apply the same sort of concept to the whole layout of the site. So let's just remind ourselves of maybe what the problem is. So if I'm looking at our album page here with quite a wide browser window, as I shrink the browser window you can see the images are getting smaller. And if I shrink it right down to the kinda size that I might be seeing on the mobile phone, you can see the images have shrunk right down.

- And the problem is, sometimes you're looking on the phone, you don't want the images to be tiny. You want to use as much of that screen as possible. And this is where responsive grid layouts come in. So let's find out how the responsive grid layout system works in Bootstrap.

So I'm gonna start with some images actually. So, let's get a square like that. So this is the basics of how the grid system works. You have eight, and one more. All right, so we have one, two, three, four, five, six, seven, eight, nine, ten, eleven, twelve columns. And okay, so imagine our layout of our page looks a bit like this. Well, we've got this sort of grid. And the concept is that I might want to take an image and I just want the image, and you decide how much of the page you want your image to take up.



This is how the page might look when someone is viewing it in nice wide desktop display. Okay, I want my images to look like this. So, maybe I want one, two, three, four, five, six images across the page.



Or maybe you want. You want take up more space and you want to say, have. Just four images across the page. And then you might go like this. Two. Three. Four.



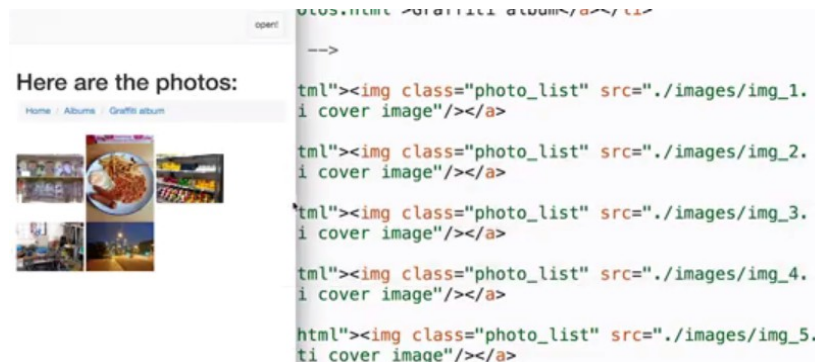
So the first step is to really decide how you want your page to lay out.

- And you can see that in this case, each image takes up three of those kind of spaces of my 12 grid columns.

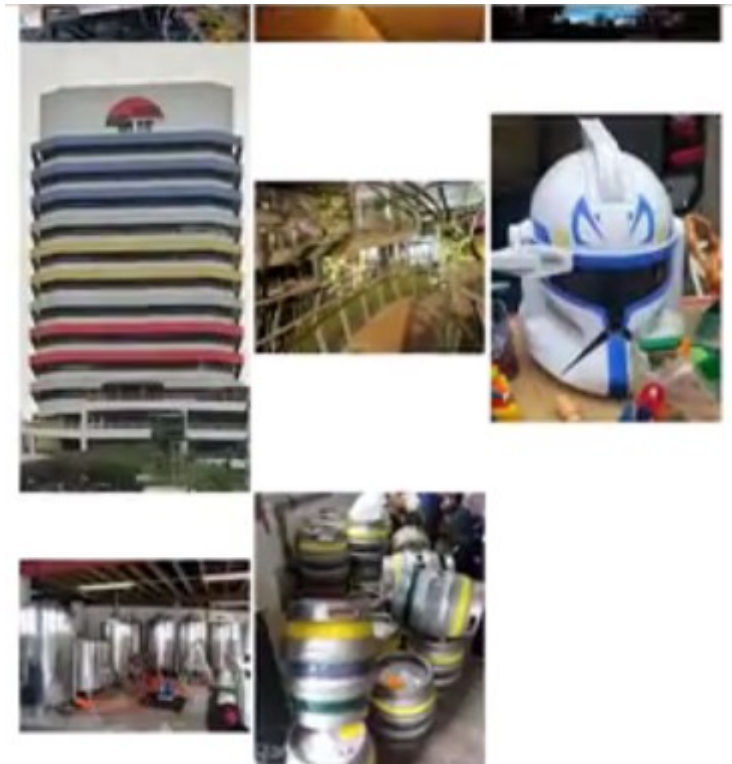
And, but then, the next step is, okay, what happens if someone's looking at this on a mobile device? If it's a mobile device, remember, it's going to be a narrow screen. And, I don't want them to see four images laid out on the narrow screen. So, in that case, you might say, well. If it's mobile I want them to be stretched out. So I want the images to take up say, six spaces on the grid like that. Obviously they're gonna be square images. So that would be going down like that, but we'll just keep it like that so we can see the grid.



So in that case, each image is now taking up six spaces on the grid. So that's how it looks visually, but now how do we turn that into code? So let's see if we can apply this kind of layout concept to our image gallery webpage. So I'm gonna open up the list of photos. Okay? So, if we view this in the browser, it looks a bit like this.



And, first thing I want to do is add a few more photos, so we can really see how that grid looks. So, lets just drop in. So, we got five so far. Six, seven, eight, nine. Actually, I'll just do it. There's an easier way of doing it when I just grab three at a time and then pop those in. Three more, three more. Now just put the numbers up. Five, six, seven, eight, nine, ten, eleven. Right, so I've now got quite a few images on this page. I've got eleven images. Save that, reload.



Okay, so now you can see where the image is, kind of sitting there like that. And that's okay. It doesn't look too bad, the layout. But as I said, we want to control exactly how it's laid out on different sized screens. So, into the code. Well, we want to, let's imagine we're looking at it in the widescreen, and we want each image to take up three columns. First thing I have to do is create the grid structure in Bootstrap. So three columns and basically what I do is put them all in a row. And like that. So now I'm thinking, yeah, I should probably remember that, that is the end of the container. End of container. And this is the end of the row. Okay, and what I want to do is say well, each of these images is gonna take up three columns, so I put each image in a div like this. And I set a special class on the div which is called

- **col-md-3**. So that says, when you're on a medium screen, whatever's inside this div will take up three columns on that screen.

```
<div class="row">
  <div class="col-md-3">
    <a href="photo1.html"></a>
  </div>

  <div class="col-md-3">
    <a href="photo1.html"></a>
  </div>

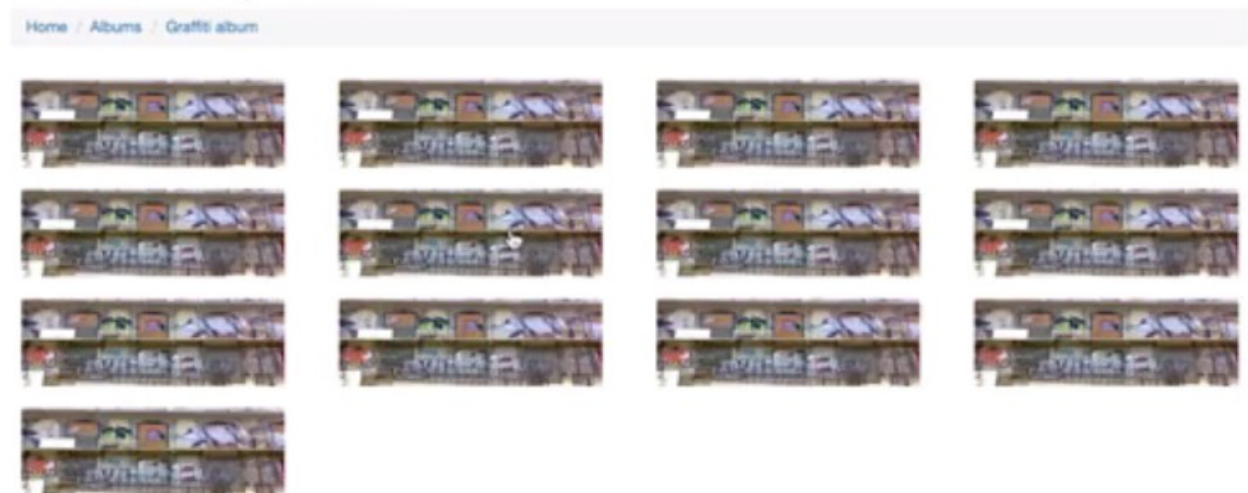
  <div class="col-md-3">
    <a href="photo1.html"></a>
  </div>
</div> <!-- / row -->
```

Okay. And let's just grab that, put that in, apply that to all the images. In fact, I'm gonna just copy paste because I'm lazy and get the end of the row. Yeah, I've got the end of the row there. So let's just copy paste that. We can use the same image for now. We can always fix it up later. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, okay, that's all right, and reload. Okay, so that's good. That's means, that's how it looks. Let's just check the code. So all I've done is I've put everything inside a row, and then each image is then put inside col-md-3. And that means that, so now each image is given enough space to take up one quarter of the complete size, and what happens if I shrink it down? Did you see that?

All right, it kind of pops down to a single row, and that's interesting, but I also notice something else, which is that the image is too wide, and that's because of the class, right? So there's a photo_list class, and so each, let's look at the photo_list class. Well, that's setting it to 25%, I want to be 100%. I want the photo to take up the whole of this little div that it sits inside. Which means, like that. Okay, and, well the problem is they're quite high now. So max-height. Maybe we can set the max-height to something to 100 pixels or something like that, right? Okay, and, Okay, and so you can see I've got that image repeated every time there. Maybe I'll just add a little bit of padding onto the image just to make it really clear adding ten pixels just so we can see exactly what's going on.

```
.photo_list{  
  width:100%;  
  max-height:100px;  
  padding:10px;  
}
```

Here are the photos:



So there's my sorta image grid and so you can see that each image is now taking up a quarter of that complete grid which is exactly what we expected from here. Now let's do a little experiment. What happens if we wanted to now, instead of taking up a quarter, we want it to take up half of the grid?

So how many columns is it taking up, if it's taking up half the grid? Well, that's six, right? So we then just go through and change all these to sixes, right?



```

<div class="row">
  <div class="col-md-6">
    <a href="photo1.html"></a>
  </div>

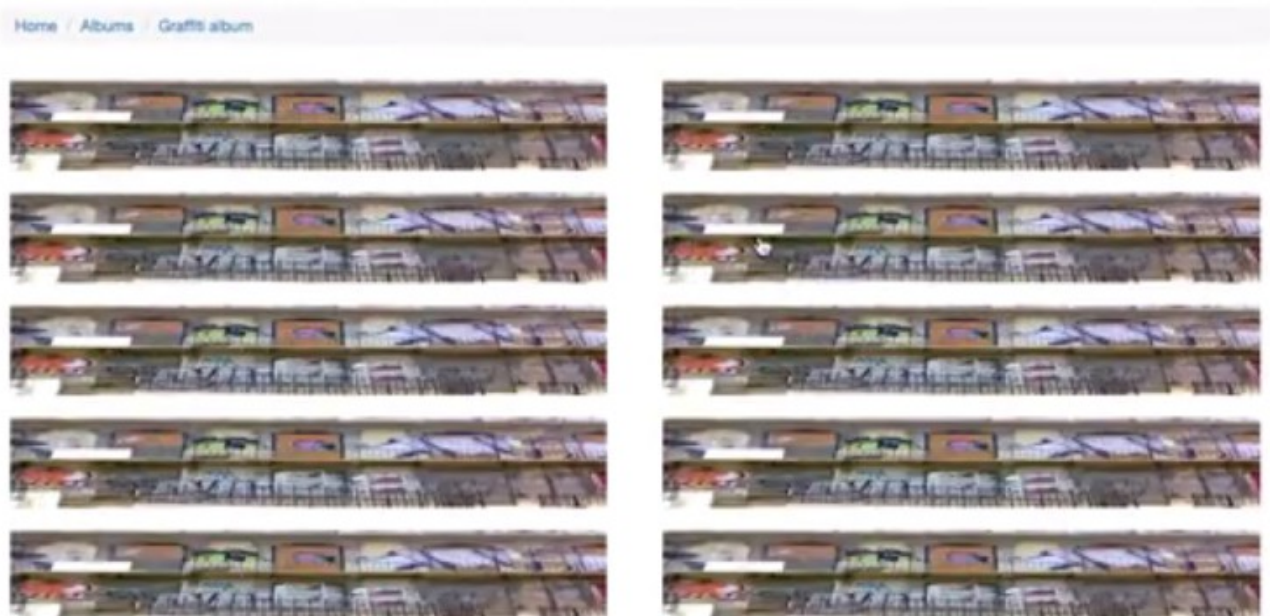
  <div class="col-md-6">
    <a href="photo1.html"></a>
  </div>

  <div class="col-md-6">
    <a href="photo1.html"></a>
  </div>
</div> <!-- / row -->

```

Just do that and I can just copy paste them. And, three, four, five, six, seven, as many as I like to illustrate it. Okay so you can see now it takes up six columns, and that's fine. That's what we expect.

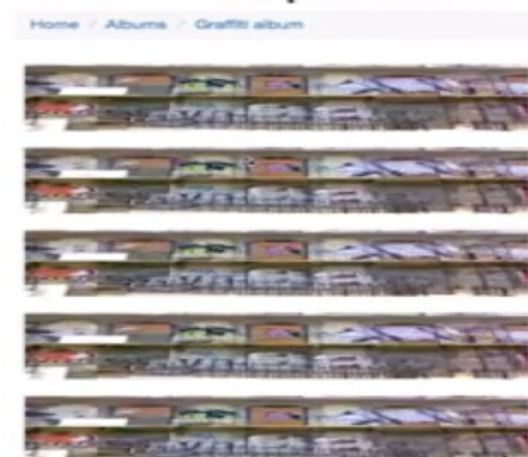
Here are the photos:



So now we're gaining control over exactly how much of the page the different images are taking up.

Okay, so that's good, but what about if the page shrinks and I'm looking at it on a mobile device. Right, you can see that it's kind of popping down to being just a single column.

Here are the photos:



So effectively what's happening is when I shrink it it's just going like that, and each image is taking up the full 12 columns. And that's okay, that's pretty decent. But what if actually I want to have a bit more control? So I say, okay, well when it's a big screen I wanted to take up the, you know, back to the three columns per image. And if it's a small screen, I want to do a different layout. So I want it to do this. I want to squeeze that out so that it looks like that. And this is quite a common way of doing it, to do is what we call a wire frame. You can draw it by hand or you can use some sort of technical kinda drawing program like this.

And so they now have defined my two layouts. So that's gonna be how many columns, so that's six, so that's taking up six, and then that's taking up three, okay? So how do we define those two different modes? Well, it's actually surprisingly easy. You go [col-xs-6](#) and [col-md-3](#), so we can just pop that in there. Okay. All right, so let's look at that code a bit. So I've now got two classes. I've got a [col-xs-6](#) and [col-md-3](#), and what that says is when they're looking at it on an extra-small screen, in other words, a mobile screen, want this particular element of the layout to take up 6 of those 12 columns.

```
<div class="row">
  <div class="col-xs-6 col-md-3 col-lg-2">
    <a href="photo1.html"></a>
  </div>

  <div class="col-xs-6 col-md-3 col-lg-2">
    <a href="photo1.html"></a>
  </div>

  <div class="col-xs-6 col-md-3 col-lg-2">
    <a href="photo1.html"></a>
  </div>
</div> <!-- / row -->
```

And when they're looking at it on medium or greater screen, I want it to take up 3. So let's see how that works. So I've reloaded. I'm looking at it on that small screen and as we pop it out, you'll see suddenly, bing it pops out and I've got four. So it kind of stretches up to the medium size layout. And then I get four. But what happens if you want to lay it out on a really big screen? So you want to define a sort of third layout if you like. Let's just first decide what that layout's gonna be.

And in this case I'm going to put six images across the screen. You can see that basically six images takes up two grid slots per image. So, then we go in and just add a final class, which is [col-lg-2](#).

And, I can then just drop that in to my classes. So, copy paste that, col-lg 2. And, we'll see how that works shortly. Right, so there we go. That's the large screen. And then you'll see, popping down to the medium screen. And then popping down to the small screen and below. So now, we've got three different layouts, depending on what size of screen it is.

- So, it's really important to kinda start your planning either as a drawing or as a technical drawing program and then it's really easy to calculate what kind of classes to put in.

So, the medium size classes, this md thing and you can dictate how many columns to take up on a medium screen then access allows, say, how many columns to take up on a small screen. And lg allows you to say how many columns to take up on a large screen. Okay, so that's really just a big run through of how responsive grids really work in Bootstrap. And hopefully that's shown you how you can create these webpages that can really respond to different types of screens. And really make the layout as usable as possible, making maximum use of the available screen size.