Exercise: Applying a Media Query to a Grid Layout

Now that you're familiar with media query syntax, let's apply this to our grid layout using a desktop-first approach.

Let's add a media query for a tablet device to use the following: Each box in this layout is 19% wide, and the margin between the boxes is 8%.

For the tablet layout, we'd like the four horizontal boxes in the desktop layout to rearrange and to add two lines of two boxes each when we reach 500 pixels, and display that way at no more than 500 pixels. The space between the boxes should still be 8%. For this breakpoint, we'd also like the background color of the boxes in the layout to change to "lightyellow."

Instructions:

1. Click the 'run code' button to view the desktop version of the layout
2. Add the appropriate media query breakpoint syntax to the CSS stylesheet. Remember to include the appropriate flex-basis and background properties for this layout described above
3. Click the 'run code' button

Run Code | Save | Export | Reset

index.html   styles.css

```
<!doctype html>
<html>
 <head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="/styles.css" />
</head>
 <body>
<div class="row">
  <div class="box">Box 1</div>
  <div class="box">Box 2</div>
  <div class="box">Box 3</div>
  <div class="box">Box 4</div>
</div>
</body>
</html>
```

Theme Select:

rubyblue

Pro tip: Remember to Resize Browser Windows to View Changes

If you don't see the changes you just coded, don't panic! How big is your device? If it's larger than 500px — and if you're on a desktop computer, that's likely true — then you'll need to make your browser window smaller to see the change.

Click on the following tab to check your answer.

Check Your Answer

To add the media query for tablet layout we need to do two things.

1. Calculate the appropriate Flexbox code
2. Add the background color (lightyellow)

To calculate the Flexbox code: for the tablet display we want to have 2 rows of 2 boxes with margin in between the boxes equal to 8%. Therefore, to calculate the flex container for the boxes in this layout:

The width of the entire screen is 100%.

To account for the margin, we subtract 8% from 100% (100 minus 8 equals 92).

Therefore, the space the two boxes can occupy is equal to 92% of the screen width. So the space each box can occupy is 46% (92 divided by 2 equals 46).

For this media query, the flex-basis property is: **flex-basis: 46%;**

** Remember that Flexbox inherently accounts for the 8% margin

The background color at this screen dimension is lightyellow so this style also needs to be added to the media query: **background-color: lightyellow;**

The media query code for the tablet layout is below.

```
@media (max-width: 500px) {
   .box {
     flex-basis: 46%;
     background-color: lightyellow;
   }
 }
```

Run Code Save Export Reset

index.html styles.css

Theme Select:

rubyblue

Add a Media Query for Mobile Devices

Now, let's apply a mobile style as well. For this layout, we'd like the boxes to be stacked on top of each other from bottom to top, at screen sizes no wider than 325 pixels. For this breakpoint, we'd also like the background color of the boxes in the layout to change to "lightgreen."

Instructions:

1. Click the 'run code' button to view the layout
2. Add the appropriate media query breakpoint syntax to the CSS stylesheet. Remember to include the appropriate flex-basis and background properties for this layout described above
3. Click the 'run code' button

Remember to resize your browser to see the effects!

Run Code Save Export Reset

index.html styles.css

```
<!doctype html>
<html>
 <head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="/styles.css" />
</head>
 <body>
<div class="row">
  <div class="box">Box 1</div>
  <div class="box">Box 2</div>
  <div class="box">Box 3</div>
  <div class="box">Box 4</div>
</div>
```

```
</body>
</html>
```

Theme Select:

rubyblue

Click on the following tab to check your answer.

Check Your Answer

To add the media query for tablet layout we need to do two things.

1. Calculate the appropriate Flexbox code
2. Add the background color (lightgreen)

There are two possible answers to this problem. One would be to keep using Flexbox and change the flex-basis to 100%. The media query code for this is below.

```
   @media (max-width: 325px) {
      .box {
        flex-basis: 100%;
        background-color: lightgreen;
      }
    }
```

A second solution would be simply to disable Flexbox. The default behavior here is to stack these boxes on top of each other, so by setting display: block; we are returning the page to its default behavior. The code for this alternative method is below.

```
 /* alternative answer
   @media (max-width: 325px) {
      .row {
        display: block;
      }

      .box {
        background-color: lightgreen;
      }
    } */
```

As the background color at this screen dimension is lightgreen, this style also needs to be added to the media query: **background-color: lightgreen;**

Theme Select:

rubyblue