

6E Layouts - Extra space Lab

In the last section we learned how to apportion the extra space when a widget needs that. Let's exercise that with a new widget - the Tickets stateful widget. But first let's warm up with our date picker.

PickDate

Right now the days of the week are one on top of another. They really need to be side-by-side. Let's put them in a row.

1. Edit `date_picker.dart`. Change the Column to a Wrap. (Remember, `Wrap()` will allow its contents to be side-by-side but wrap to the next line if it's too wide).

Notice that everything is bunched to the left.

2. Change the `wrapAlignment` to a couple of different values. Pick the one you like the best.



Checkout

The expiry month and expiry year fields are in a `Row()` but they're too close to each other. They feel crowded, don't they?

3. Put a `Spacer()` between them.
4. Run and test.

Better, but the space is too big, right? Why? (Hint: the horizontal width is distributed equally among the `Row()`'s three children.) We need to give way more width to the fields and way less to the `Spacer()` between them.

5. Add flex to each of the three:

`Flexible(`

```

flex: 2, // <-- Add this
child: DropdownButtonFormField(
  ...
),
),

```

6. A flex of 2 is probably not the right number. Experiment with flex factors on all three until you like the ratio.

Getting the user's tickets

You've got a ticket.dart StatefulWidget but it's a little bit of a mess. Let's clean it up a little here and in the next few labs.

7. Notice that when you have multiple tickets, it's hard to tell where one ends and the next begins.
8. Put a SizedBox() widget between each ticket. You can determine how much height you think is appropriate.

Bonus!! Getting a QR code

Only if you have extra time, put a QR code on each ticket. Here's the idea. Our customer buys one or more seats. When they physically show up at the theater our employees will show them to their seat. It would be kind of cool for them to show their phone and have our employee scan a QR code to 1) register the customer as having shown up and 2) to know where to seat them. In this bonus, let's show that QR code on the ticket itself.

9. Go to pub.dev and search for a QR code creator. Make note of your favorite.
10. Add it to pubspec.yaml.

When you save, VS Code should automatically download and install your chosen package.

11. Edit ticket.dart. Add an import to the top of the file.
12. Figure out how to call the library methods to print on each ticket a QR code instead of the ticket number.
13. Run and test. Every phone has a built-in QR scanner. Use your phone to scan the QR code. Does the data make sense?