

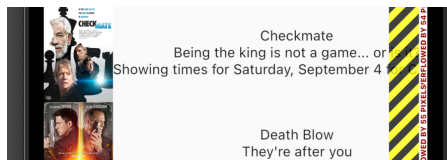
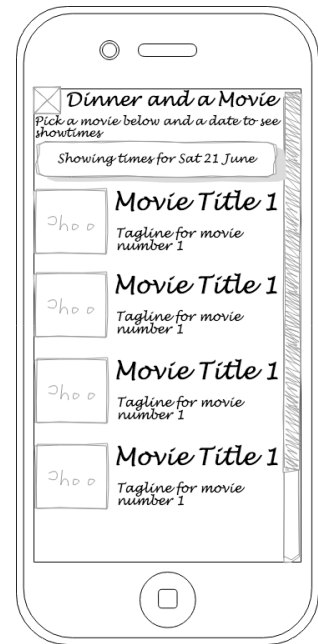
6C Placing widgets (next to or below)

Look at the wireframe of the Landing scene. Focus at the top. It looks like our clients want their logo to the left and the business name (Dinner and a Movie) to the right. We can do that!

1. Edit Landing. You currently have a Column. That's fine except that you can't get things to be side-by-side in a Column(), so you'll need a Row(). The 'cell' to the left should contain the logo and the 'cell' to the right will be your Text(). Go ahead and make that happen.
2. Run and test. If they're side-by-side, let's move on to another challenge.

If you glance back at the wireframe of Landing, you may notice that in each FilmBrief the poster is to the left of the film's title and tagline.

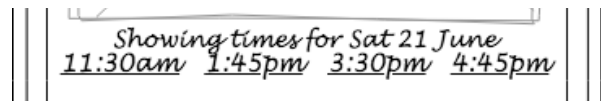
3. Edit FilmBrief. You currently have a Column. To get the poster to the left, you'll need a Row(). The cell to the left should contain the poster and the cell to the right will be your existing Column() with the title and tagline above-and-beneath.



4. Run and test. Make sure it works but note that you may have too little room for the poster and title and tagline. Don't worry about that right now. We'll fix that soon enough.

Showing Times

5. We want the times to be side-by-side as well. Put them all in a Row() instead of a Column(). They'll be bunched up and again, it may or may not overflow. Don't be concerned with that yet.



Making FilmDetails responsive with Flex

In FilmDetails, you have the poster above the details in a column. That looks really good on a device that is in landscape orientation but in portrait, it's not great because the poster is just too big. It would look nice to have the poster on the left and all the details on the right.

6. Put in a Flex() with the children property set to a List of two widgets; The first being the poster and the second being your existing Column() (but without the poster. duh).

7. Hardcode the direction property to `Axis.vertical`.
8. Run and test. You should see no difference.
9. Now set direction to `Axis.horizontal` and test again. They should be side-by-side now. As before, don't sweat an overflow. We'll fix this soon enough.
10. Add a ternary to the direction property. If we're in portrait, make the direction `Axis.vertical` and if we're in landscape, make it `Axis.horizontal`. (Hint: `MediaQuery.orientationOf(context)`).
11. If you rotate your device, you'll see it responsive -- when in portrait, the poster is above the text but in landscape, the poster and text are side-by-side.

Bonus! Sizing the poster

Notice that the poster is too big. Let's fix that real quick.

12. Add an orientation-aware height and a width to make the poster look a little better. Here's a hint:

```
Image.network(  
  "http://localhost:3008/${selectedFilm.posterPath}",  
  fit: BoxFit.cover,  
  height:  
    isLandscape ? MediaQuery.sizeOf(context).height * 0.8 : null,  
  width: isPortrait ? 300.0 : null,  
)
```

Obviously you'll need to create the `isLandscape` and `isPortrait` variables based on the current orientation. We'll leave that as an exercise to you.

13. Give it a run. Notice how well-behaved the poster is when you rotate the device.

Way to go! You're finished.