Responsive Design Lab

If you look at your pages currently, they're pretty tolerant of different screen sizes. For instance, the film cards will go down to just one column on a small screen but will be side-by-side-by-side on a widescreen monitor. All of the lists of things that are side-by-side will wrap on a narrow screen. And even the top menu changes size with the screen size.

But sometimes we want to literally change the way something is laid out when the screen size changes. We want to hide things, change the order of things, or move things around. This is where responsive design comes in to play.

Prepping for responsive development

1. Let's do an experiment. Add this to the bottom of site.css:

```
@media (max-width:768px) {
  body {
    background-color: yellow;
  }
}
```

2. Browse to any page on your site and narrow the window to less than 768 pixels. You should see the background change to yellow when you do. This is a trick developers use while testing so we can instantly see where the break is. Leave that in until the end of the lab if you like. You'll want to remove it eventually.

Hiding some days

On the landing page, you have a list of days of the week. They wrap when the screen is narrow but they don't wrap very attractively. Let's just hide the days at the end of the list.

3. Edit landing.html and add a class to the days at the end of the list. How many days? That can be up to you. Start with two and add/remove as you desire.
4. Edit site.css. Finding the media query that you added above, add to it a style that sets display: none; to those days.
5. Run and test. When the page gets narrow, those days should just not appear.

Changing a layout

6. Look at detail.html in a browser. It looks great on a laptop but when the screen width gets smaller, the layout starts to look clunky and we're forced to side-scroll. Let's put the poster and film info above and beneath.
7. If you don't already have detail.html laid out as flexbox, please do so now. Remember, you'll set display: flex; on the container of #poster and #info.
8. Now, to make them flex with the screen size. It'll be super … well … *flexible*.
9. In your media query, set flex-direction: column; for the container.

10. Run and test. They should change based on the width of the screen.
11. One more trick … Suppose we wanted the poster at the top and the info below it. Try changing the flex-direction to column-reverse and test it again.

12. Last step. Don’t forget to remove that yellow background-color we were using for debugging.

Congrats! You have a responsive site. There are lots of other things we could do to our site, but this will suffice for learning purposes.