How to display images

We learned that images in React can be bundled or loaded live. Let's look at both in this lab. We'll start with loaded live.

Images loaded via http requests

1. Edit LandingPage.js. You'll notice that we have one <img> tag where the movie poster is supposed to be displayed. We're going to hardcode that value for now. But to what value?
   a) src="c:\myproject\client\public\img\posters\1.jpg"
   b) src="../public/img/posters/1.jpg"
   c) src="/img/posters/1.jpg"

The answer is "c", of course. But we include those first two because they're often mistakes that developers make. Hopefully seeing the wrong answers will remind you of how not to do it.

2. Set its src property to the right value.
3. View LandingPage in the browser to make sure you can see a poster. Don't worry about the size yet. We'll fix that in a later lab.
4. Do the same for the poster in FilmBrief.

Embedding images

A movie poster is a perfect candidate for an on-demand image; it is huge and it is likely to change between compiles of your React application. But if you have a small image and that image won't change, you might embed it in the bundle. Let's practice that next.

In the PickSeats component we'll eventually have a map of the theater so the user can tap/click on a seat to reserve it. We'll need to display a bunch of seats and fortunately we can re-use the same small image over and over. It'll be to our benefit to embed that image.

5. Make PickSeats your startup component. Go ahead and view it in the browser.
6. In the /public/img folder, you'll see there's a seat.png file. Take a look at it if you like.

Everything that is to be compiled in the bundle must be located under src. So we'll have to move or copy seat.png to that folder.

7. Create a folder under src called bundledImages.
8. Copy or move seat.png to that folder.
9. Edit PickSeats.js. Find where it says "SEATS WILL GO HERE" or something like that. Replace that whole <p> with this:

```
<div style={styles.seatWrapper}>
  <div style={{...styles.seatItself }} >
      #
  </div>
</div>
```

10. Change your style to look like this:

```
const styles = {
  seatWrapper: {
    margin: "5px",
  },
  seatItself: {
    backgroundImage: `url(${seatImage})`,
    backgroundSize: "100% 100%",
    width: "30px",
    height: "30px",
    fontWeight: "bold",
    display: "flex",
    alignItems: "center",
    justifyContent: "center",
    borderRadius: "10px 10px 0px 0px",
  },
}
```

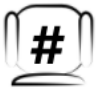
Don’t worry about what all that means, the React syntax will become clear enough in the next few chapters.

If you run and test this now, it complains that it can’t find a variable called seatImage. That variable is supposed to be populated with the base64-encoded image of the seat. You’ll bring that in by importing the seat image file.

11. Add this import to the top of the component:

```
import * as seatImage from './bundledImages/seat.png';
```

That’ll solve the error and in the browser you should see the seat image like this:



Did you notice that “#” in the seat? That’s the text from the <div> you placed a few steps earlier. We’ll make that number be dynamic eventually so the user can see a seat number on the map.