Repetition: Fragments

In earlier versions of this course, this section also introduced the concept of "**React Fragments**" (`<Fragment> ... </Fragment>` or `<> ... </>`).

The newer version of the course already introduced this concept in the "React Essentials" sections.

But since it's a key concept that will be used throughout the entire course (and, in general, in pretty much all React projects), it's time for a brief refresher!

When writing JSX code, there's one important rule: A JSX value must have only one root element.

For example, the following code would be invalid and cause an error:

```
return (
  <h2>Welcome!</h2>
  <p>React is awesome!</p>
);
```

So would this code:

```
const content = (
  <h2>Welcome!</h2>
  <p>React is awesome!</p>
);
```

In both snippets, the JSX value has two sibling root elements - and that's not allowed!

One solution would be to wrap these elements into a `<div>` - which then acts as a single root JSX element:

```
return (
  <div>
    <h2>Welcome!</h2>
    <p>React is awesome!</p>
  </div>
);
```

This would work and therefore is an acceptable solution.

But it has a downside: You now have that extra `<div>` in your DOM - even though you don't really need it (besides for getting rid of the this error).

That's why React offers a better solution: A special JSX element called "**React Fragment**".

It can be used as a wrapper to ensure that there's only one root JSX element whilst at the same time not rendering any DOM element.

You can use it like this:

```
import { Fragment } from 'react';

// ... other code ...

return (
  <Fragment>
    <h2>Welcome!</h2>
    <p>React is awesome!</p>
  </Fragment>
);
```

Most React projects (e.g., projects created with Vite or create-react-app) offer an even shorter form:

```
// no import needed

return (
  <>
    <h2>Welcome!</h2>
    <p>React is awesome!</p>
  </>
);
```