**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:

1. return (
2. <h2>Welcome!</h2>
3. <p>React is awesome!</p>
4. );

So would this code:

1. const content = (
2. <h2>Welcome!</h2>
3. <p>React is awesome!</p>
4. );

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:

1. return (
2. <div>
3. <h2>Welcome!</h2>
4. <p>React is awesome!</p>
5. </div>
6. );

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:

1. import { Fragment } from 'react';
3. // ... other code ...
5. return (
6. <Fragment>
7. <h2>Welcome!</h2>
8. <p>React is awesome!</p>
9. </Fragment>
10. );

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

1. // no import needed
3. return (
4. <>
5. <h2>Welcome!</h2>
6. <p>React is awesome!</p>
7. </>
8. );