**Learn React: How to export, import, and render a component inside a component -** [Aphinya Dechalert](https://medium.com/@PurpleGreenLemon?source=post_page-----5d432f1171ab--------------------------------) Jan 2, 2022

Part 5 of Learn React from Scratch Series

Previously on ***Learn React from Scratch Series*:**

In React, a component can consist of the smallest possible UI element or a collection of UIs. For a collection of UIs, it requires rendering components inside a component.

For example:

A picture containing diagram

Description automatically generated

If we look at a view, a UI can contain multiple repeating elements grouped together. For example, a navigation bar often contains a set of menu links and a search bar. The elements in this UI group are often used together.

Here is a code sample on how to render it in React:

class Logo extends React.Component {  
 render(){  
 return (  
 <div className="logo">  
 <img src="/squirrelLogo.jpg">  
 </div>  
 );  
 }  
 }  
 ​  
 class Navigation extends React.Component {  
 render(){  
 return (  
 <div className="navigation">  
 <Logo />  
 <!-- some code here -->  
 </div>  
 );  
 }  
 }

Most of the time, we usually don’t have everything in a single file. When this happens, you can use a import statement to bring the definition into context. import is not specific to React but is part of JavaScript's module system. The module system lets you keep your code independent from each other until needed.

Here is an example of a import statement:

import { Logo } from './Logo';

By default, if there is no file extension, then .js is always assumed.

To make a class import work, the target class also needs to be exported. This is called a "named export" and will also work with any top-level var, let, const, function, or class.

All you have to do is place the keyword export in front of the *thing* you want to export.

Here is a syntax example:

export class Logo extends React.Component {  
 render(){  
 return (  
 <div className="logo">  
 <img src="/squirrelLogo.jpg">  
 </div>  
 );  
 }  
 }

**React FAQ: Why do some**import**have**{}**while others don't?**

This is because there are two ways to export your top-level types.

If we see an export with the curly braces {}, then it means that the component exported is a default export. This is the only export of its kind within your entire app. The export is also written like this:

class ComponentName extend React.Component {  
 ...  
 }  
 ​  
 export default ComponentName;

When there are {} involved in the import, then it means that there may be a component with the same name elsewhere in your app. This can be a problem if you're using both of the same-named items in your component. The perk of {} is that it lets you use an alias.

For example, here is the syntax for a named export:

export class ComponentName extends React.Component {  
 ...  
 }

Here is how you can import from duplicate named exports using aliases:

import { ComponentName as DuplicateOne} from './ComponentName';  
 import { ComponentName as DuplicateTwo} from './ComponentNameAgain';