Modularity & Import/Export in React

This lesson explains how modules are created and accessed in JavaScript.

What are Modules?

It is always recommended to define only one class per file to make it strongly cohesive. These files are basically *modules*. So, if there are two classes Person and Student defined in the same file, you can split them into two modules and name them as `Person.js` and `Student.js`. Note that you must place these files inside your `App` directory.

> **Did you know?**
>
> *The classes defined in modules are private by default and can not be accessed by other files that exist inside your project. You can make them public by using import/export statements.*

Fortunately, the JavaScript community settled on one way to import and export functionalities from files with JavaScript ES6 with import and export statements.

However, being new to React and JavaScript ES6, these import and export statements are just another topic which requires explanation when getting started with your first React application. Pretty early you will have your first imports for CSS, SVG or other JavaScript files. The create-react-app project already starts with those import statements:

```
import React, { Component } from 'react';
import logo from './logo.svg';
import './App.css';
import './App.css';
```

It’s great for the starter project, because it offers you a well-rounded experience how other files can be imported and (exported). Also the App component gets imported in the src/index.js file. However, when doing your first steps in React, I try to avoid those imports in the beginning. Instead, I try to focus on JSX and React components. Only later the import and export statements are introduced when separating the first React component or JavaScript function in another file.

So how do these import and export statements work? Look at the last line of `myfile.js` to see how the variables `firstname` and `lastname` are exported from `myfile.js` to another. They can then be imported in another file with a relative path to the first file as per the following:

```
const firstname = 'Robin';
const lastname = 'Wieruch';

export { firstname, lastname };
```

So, it’s not necessarily about importing/exporting components or functions, it’s about sharing everything that is assignable to a variable (leaving out CSS or SVG imports/exports, but speaking only about JS). You can also import all exported variables from another file as one object:

```
const firstname = 'Robin';
const lastname = 'Wieruch';

export { firstname, lastname };
```

Imports can have an alias. It can happen that you import functionalities from multiple files that have the same named export. That’s why you can use an alias:

```
const firstname = 'Robin';
const lastname = 'Wieruch';

export { firstname, lastname };
```