Setting up the create-react-app

In this lesson, we Install node and npm to set up a create-react-app project and have a look at a React class component

Installing node

If you are developing with JavaScript, there is no way around node and npm. You can find the installation instructions on the official website. Installing node will install `npm` (node package manager) as well. The node package manager is used to install libraries/frameworks (node packages) using the command line for your project whereas the 'node' command itself can be used to compile and run a JS file in a terminal, i.e.

```
$ node example.js
```

There exists an alternative to install and manage node: nvm. You should be able to install it via Homebrew for MacOS, and there exists a Windows version for nvm too. I highly recommend using nvm, because you can install and manage multiple node versions with it. Furthermore, it helps you to install your node version as well.

Starting a create-react-app Project

Follow the official create-react-app setup guide and read the file `App.js` which contains skeleton code for a React App. It should look something like the following.

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

```
class App extends Component {
  render() {
    return (

      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <h1 className="App-title">Welcome to React</h1>
        </header>
        <p className="App-intro">
          To get started, edit <code>src/App.js</code> and save to reload.
        </p>
      </div>
    );
  }
}

export default App;
```

As you can see this *React class component* might be quite the enigma for someone who does not have much experience with JavaScript even though it is the barebones of a React project. Arguably, a React class component may not be the best starting point. There is loads of non-React-specific information to digest for newcomers: class statements, class methods and inheritance.

Also, the import statements are only adding complexity when learning React. So, the following lessons will shed some light on the fundamental JavaScript building blocks that you would need to understand code like this without worrying about React. We will revisit this component at the end of this course, and by that point, hopefully, it will be easy to understand. In the next lesson, we'll study the types of variable declarations in JavaScript.