ES6 Destructuring

A brief introduction to destructuring and things we can access through it.

Destructuring in JavaScript ES6 provides easier access to properties in objects and arrays. Compare the following snippet in JavaScript ES5 and ES6:

```
// comment either ES5 or ES6 code

const user = {
  firstname: 'Robin',
  lastname: 'Wieruch',
};

// ES5
// var firstname = user.firstname;
// var lastname = user.lastname;

// ES6
var { firstname, lastname } = user;

console.log(firstname + ' ' + lastname);
```

While we add an extra line each time we access an object property in JavaScript ES5, it takes just one line in JavaScript ES6. For readability, use multi lines when you destructure an object into multiple properties.

```
var {
  firstname,
  lastname
} = user;
```

The same concept applies to arrays. You can destructure them, too, again using multilines to keep your code scannable and readable.

```
var users = ['Robin', 'Andrew', 'Dan'];

var [
  userOne,
  userTwo,
  userThree
] = users;

console.log(userOne, userTwo, userThree);
```

Note that the local state object in the App component can get destructured the same way. You can shorten the filter and map line of code.

```
  render() {
    const { searchTerm, list } = this.state;
    return (
      <div className="App">
        ...
        {list.filter(isSearched(searchTerm)).map(item =>
          ...
        )}
      </div>
    );
```

You can do it the ES5 or ES6 way:

```
// ES5
var searchTerm = this.state.searchTerm;
var list = this.state.list;

// ES6
var { searchTerm, list } = this.state;
```

But since the course uses JavaScript ES6 most of the time, you should stick to it.

Further Reading:

- Read about ES6 destructuring