Upgrade React

Upgrading an existing React application to version 18 only requires two steps

If you are already using the latest version of `create-react-app` which uses React version 18 you can skip this section.

What is ES6?

ES6 stands for ECMAScript 6.

ECMAScript was created to standardize JavaScript, and ES6 is the 6th version of ECMAScript, it was published in 2015, and is also known as ECMAScript 2015

Why Should I Learn ES6?

React uses ES6, and you should be familiar with some of the new features like:

- Classes
- Arrow Functions
- Variables (let, const, var)
- Array Methods like `.map()`
- Destructuring
- Modules
- Ternary Operator
- Spread Operator

Classes

ES6 introduced classes.

A class is a type of function, but instead of using the keyword `function` to initiate it, we use the keyword `class`, and the properties are assigned inside a `constructor()` method.

Arrow Functions

Arrow functions allow us to write shorter function syntax

```
hello = function() {
  return "Hello World!";

}
```

Variables

Before ES6 there was only one way of defining your variables: with the `var` keyword. If you did not define them, they would be assigned to the global object. Unless you were in strict mode, then you would get an error if your variables were undefined.

Now, with ES6, there are three ways of defining your variables: `var`, `let`, and `const`.

Array Methods

There are many JavaScript array methods.

One of the most useful in React is the `.map()` array method.

The `.map()` method allows you to run a function on each item in the array, returning a new array as the result.

In React, `map()` can be used to generate lists.

Destructuring

To illustrate destructuring, we'll make a sandwich. Do you take everything out of the refrigerator to make your sandwich? No, you only take out the items you would like to use on your sandwich.

Destructuring is exactly the same. We may have an array or object that we are working with, but we only need some of the items contained in these.

Destructuring makes it easy to extract only what is needed.

Spread Operator

The JavaScript spread operator (`...`) allows us to quickly copy all or part of an existing array or object into another array or object.

Modules

JavaScript modules allow you to break up your code into separate files.

This makes it easier to maintain the code-base.

ES Modules rely on the `import` and `export` statements.

Export

You can export a function or variable from any file.

Let us create a file named `person.js`, and fill it with the things we want to export.

There are two types of exports: Named and Default.

Named Exports

You can create named exports two ways. In-line individually, or all at once at the bottom.

Default Exports

Let us create another file, named `message.js`, and use it for demonstrating default export.

You can only have one default export in a file