ES6 Classes

This lesson explains JavaScript classes and how to define them in your application.

JavaScript ES6 introduced classes, which are commonly used in object-oriented programming languages. JavaScript, always flexible in its programming paradigms, allows functional programming and object-oriented programming to work side-by-side.

While React embraces functional programming, e.g. immutable data structures and function compositions, classes are used to declare ES6 class components. React mixes the good parts of both programming paradigms.

Consider the following Developer class to examine a JavaScript ES6 class without a component.

```
class Developer {
  constructor(firstname, lastname) {
    this.firstname = firstname;
    this.lastname = lastname;
  }

  getName() {
    return this.firstname + ' ' + this.lastname;
  }
}
```

A class has a constructor to make it instantiable. The constructor takes arguments and assigns them to the class instance. A class can also define functions. Because the function is associated with a class, it is called a method, or a class method.

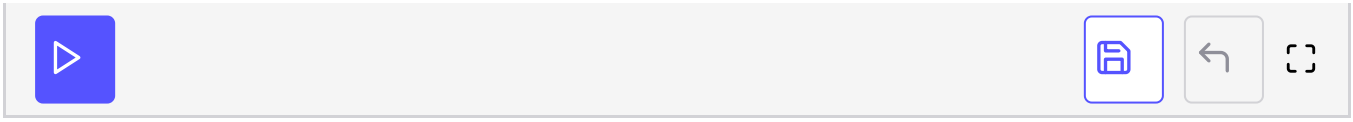
The Developer class is only the class declaration we use here, as you can create multiple instances of a class by invoking it. It is similar to the ES6 class component, which has a declaration, but you have to use it somewhere else to instantiate it:

```
class Developer {
  constructor(firstname, lastname) {
    this.firstname = firstname;

    this.lastname = lastname;
  }

  getName() {
    return this.firstname + ' ' + this.lastname;
  }
}


const robin = new Developer('Robin', 'Wieruch');
console.log(robin.getName());
// output: Robin Wieruch
```



React uses JavaScript ES6 classes for ES6 class components, which you have already used at least once so far:

```
import React, { Component } from 'react';

...

class App extends Component {
  render() {
    ...
  }
}
```

`extends`

When you declare the App component it extends from another component.

In object-oriented programming, the term “extends” refers to the principle of inheritance, which means that functionality can be passed from one class to another. The App class extends from the Component class, meaning it inherits functionality from the Component class. The Component class is used to extend a basic ES6 class to an ES6 component class. It has all the functionalities that a component in React needs. The render method is one function you have already used. You will learn about other component class methods as we move along.

The `Component` class encapsulates all the implementation details of a React component, which allows developers to use classes as components in React.

Methods exposed by a React `Component` are its public interface. One of these methods must be overridden, while the others don't need to be overridden.

You will learn about these when we discuss lifecycle methods later. The `render()` method has to be overridden because it defines the output of a React `Component`, so it must be defined. These are the basics of JavaScript ES6 classes, and how they are used in React to extend them to components.

Further Readings:

- Read about JavaScript fundamentals before learning React
- Read about ES6 classes