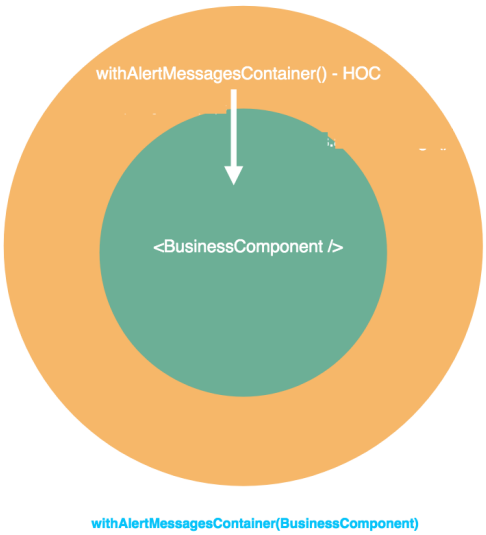
What is HOC?

We have seen the higher order function in JavaScript, which is a function that accepts another function as a parameter.

Well, React clones and imitates that functionality with a component, so it gives us the possibility to create a Higher-Order Component which is a component that accepts a component as a parameter.

Essentially, a higher-order component is a function that takes a component and returns a new component.



Why HOC?

In software development, there is one important principle that is commonly used by the name of « DRY » (which stands for "Don't Repeat Yourself").

Creating a simple utility function, that is used across several parts of the codebase is something you may have done repeatedly. You are essentially following the DRY principle by doing so. You are reusing the same utility function, without repeating the code.

In React, one of the ways to follow "DRY" across components, is to use the Higher-Order Component (HOC) concept. We can share common functionalities without repeating code.

Example of HOC

In this example, we will add a waiting message on the wrapped component. Don't get too overwhelmed by this strange code. We'll break it down later.

```
import React from "react";

const higherOrderComponent = WrappedComponent => {
 class HOC extends React.Component {
   state = {
     isLoading: false
   };

   render() {
     return this.state.isLoading ? (
       <h1>Wait a moment...</h1>
     ) : (
       <WrappedComponent />
     );
   }
 }

 return HOC;
};
```

Previous next