

EdX and its Members use cookies and other tracking technologies for performance, analytics, and marketing purposes. By using this website, you accept this use. Learn more about these technologies in the [Privacy Policy](#).



[Course](#) > [Modul...](#) > [Functio...](#) > [Functio...](#)

Audit Access Expires Apr 20, 2020

You lose all access to this course, including your progress, on Apr 20, 2020.

Functional Components

React Components

A React Component is an independent reusable component that outputs a React Element based on its properties and state.

There are two types of React Components:

- Functional Components
- Class Components

Class Components have state, lifecycle methods, and properties while Functional Components only have properties. In this module, we will cover Functional Components while Class Components will be covered in Module 2.

Functional Components

Functional Components are just functions that output React Elements. By convention, the first letter of the function name should be capitalized.

Here is an example:

```
function HelloWorld(){  
  return <h1>Hello World!</h1>  
}
```

You can use the React Component in JSX by creating an HTML tag with the same name as the React Component:

```
var element = <HelloWorld/>
```

Another Example:

```
ReactDOM.render(  
  <HelloWorld/>,  
  document.getElementById("root")  
)
```

These examples will all evaluate to the React Element that is returned by the HelloWorld Component.

Adding Properties to Functional Components

The first argument to a Functional Component is an object that contains the component's properties.

```
function HelloWorld(props){  
  return <h1>Message: {props.message}</h1>  
}
```

You can supply property values the same way as you supply attribute values:

```
ReactDOM.render(  
  <HelloWorld message="Hello World!"/>,  
  document.getElementById("root")  
)
```

Properties can be string literals, arrays or any other type of JavaScript object including other React Elements:

```
function HelloWorld(props){  
  return <h1>Value: {props.numberArray[props.index]} </h1>  
}  
  
ReactDOM.render(  
  <HelloWorld index = "3" numberArray={[1,2,3,4,5]}/>,  
  document.getElementById("root")  
)
```

You can supply as many property values as you want and they will all be accessible through the props argument.

© All Rights Reserved