

Table of Content

- Introduction to React
- JSX in React
- Rendering Elements
- Components and Props
- Components and Props 2
- State and Lifecycle

Components and Props 2

Props stand for Properties. It is used to Pass information into the React Components. For example, if we create a custom Component, then by using Props, we can pass data into it in the form of Objects, strings, numbers, etc.

Named export and Default Export:

Before learning about Props and Prop-types, let's learn the Named export and Default Export:

1. Default Export:

In this case, if import occurs by any name, Suppose XYZ, then the default value will be exported. It is generally used to export a single object, function, or variable. Let's understand with an example-

Create two files. We are creating module.js and module2.mjs. We will be performing a default export from module.js to module2.mjs

In module.js:

```
const a = "Harry";  
const b = "Rohan";
```

```
const c = "Akash";
```

In module2.js

```
import XYZ from "./module.js"  
console.log(XYZ);
```

The default exported value, which is 'Rohan', will be imported. In default Export, the name of Import is Independent of the name of Export.

2. Named Export:

In this case, we export some specific values. The name of import is dependent on the name of Export which means You can't use a different name in the Import and Export function. We use Curly brackets for Named export. For example: performing the same export using named export.

In module.js

```
const a = "Harry";  
const b = "Rohan";  
const c = "Akash";
```

In module3.js

```
import { b } from "./module.js"  
console.log(b);
```

The value of 'b' will be imported, which is 'Rohan'. Name Exports are used to export multiple values.

Changing values using props:

Now let's Suppose you want to use the above navbar in your different applications but with different titles and About, or pass New values to your existing Navbar. You can do so by using Props in React.

In navbar.js:

```
export default function Navbar(props) {
  return (
    <nav>
      <p className="title">{props.title}</p>
      <p className="about">{props.about}</p>
    </nav>
  );
}
```

Here, we are passing props in our Navbar component and have created two props: props.title, props.about

Now we will be passing values in these props.

In app.js:

Passing values: You can pass the values in the created props as shown below:

```
function app() {
  return (
    <navbar title="Textutils" about="About" />
  );
}
```

Props Types

propTypes is a feature in React that allows you to define the expected data types of the props passed to a component.

You first have to import PropTypes. In functional components, propTypes are not defined as a static property, but rather as a property of the function itself, like this:

```
import PropTypes from 'prop-types';

function MyComponent(props) {
  // component rendering logic
}

MyComponent.propTypes = {
  someProp: PropTypes.string.isRequired,
  anotherProp: PropTypes.number,
};
```

It means that my prop-type of title is a string which means on passing any other value, like Number, it will show an error in the console. Hence we can only pass a string in props.title and props.aboutText. We can use 'isRequired' keyword, which makes sure that we won't leave that prop blank. If you do so, it will show an error in the console. Similarly, you can use an object, Number, etc. prop-type as well.

Default Prop-Type

```
navbar.defaultProps = {  
  title: "Set title here",  
  about: "About text here"  
};
```

[Get Download Link](#)