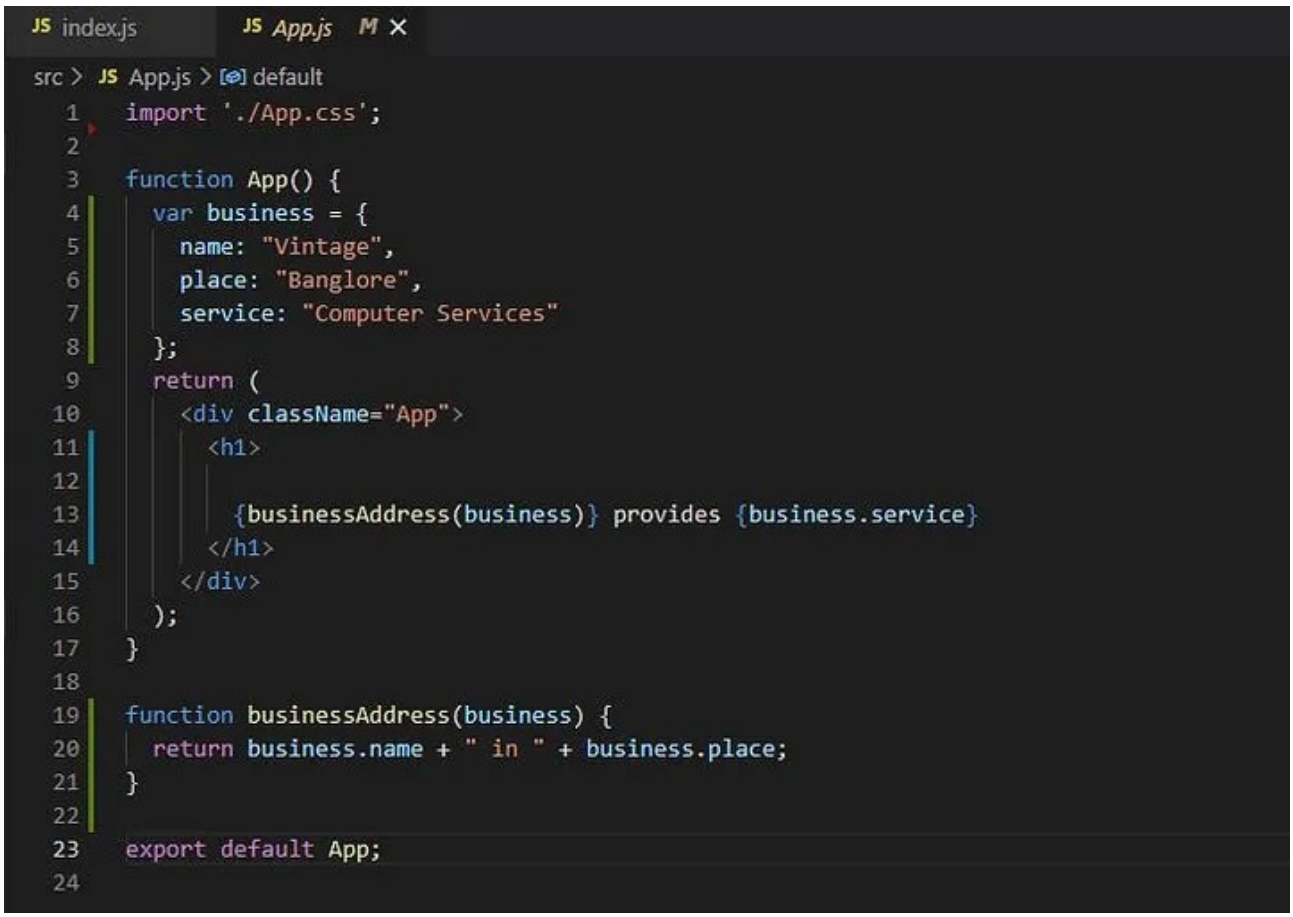
ReactJS State and Props Examples

Simple JSX Example:

object value {business.service}, function call {businessAddress(business)} or any valid JavaScript expressions are specified inside curly braces.



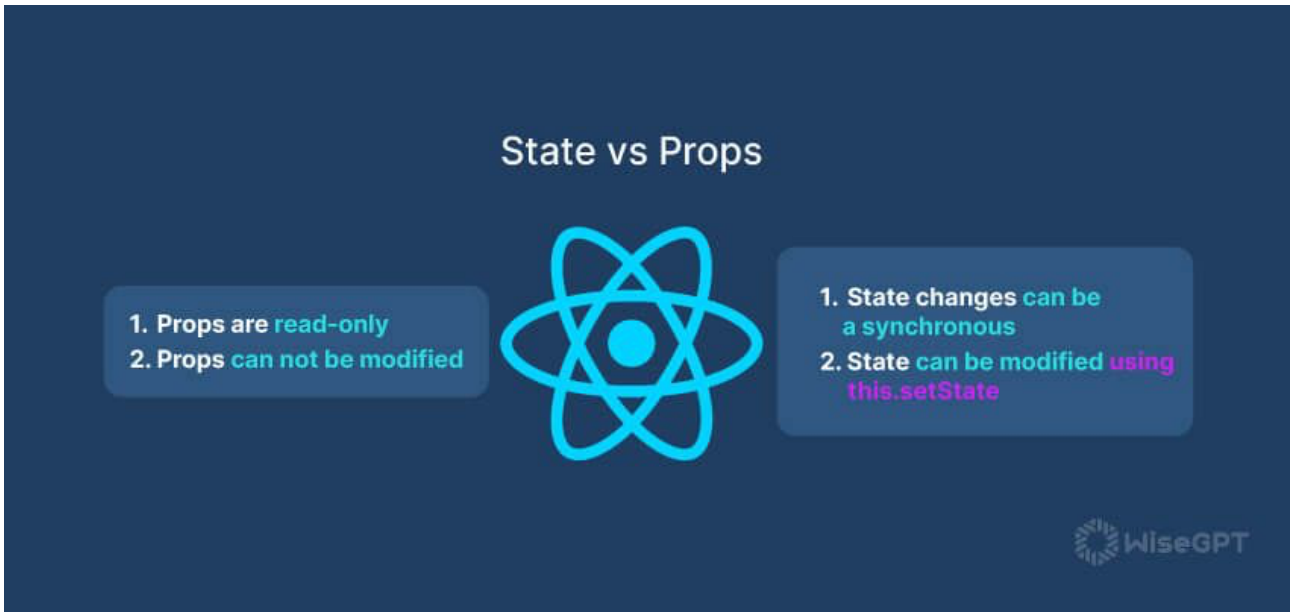
output:

ReactJS State:

- State of a React component is an object that contains information.
- State object stores values of the properties related to component.
- Hence whenever there is change in the properties related to component, the value associated with the state object changes and the component will be re-rendered.

ReactJS Props:

- Props in ReactJS are used to send data to components.
- Props are immutable, its values cannot be modified by a component that is receiving it from outside.
- Props are used either in state or functional components.
- Props are used to pass data between parent and child components.



Example for ReactJS State-

In the below example state objects are “userName” and “userPlace”. These state object values are accessed using “this” pointer. Below state is declared and used inside class component “App” which extends ReactJS default “Component” (**App.js file**).

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

export default class App extends Component {

  //State declaration
  state={
    userName:"Peter",
    userPlace:"Newyork"
  }
  render() {
    return (
      <div>
      {/* accessing state object value using "this" pointer */}
      {this.state.userName},{this.state.userPlace}
      </div>
    )
  }
}
```

Below is the **index.js** file in which **App.js** file has been rendered

App.js index.js

src > index.js

```
1   import React from 'react';
2   import ReactDOM from 'react-dom';
3   import './index.css';
4   import App from './App';
5   import reportWebVitals from './reportWebVitals';
6
7   ReactDOM.render(
8     <React.StrictMode>
9       <App />
10    </React.StrictMode>,
11    document.getElementById('root')
12  );
13
14  // If you want to start measuring performance in your app, pass a function
15  // to log results (for example: reportWebVitals(console.log))
16  // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
17  reportWebVitals();
18
```

Index File

Output:

Example for ReactJS Props

- In the below example we are passing “props” values from Parent to Child component. **Here both Parent and Child components are Class components.**

Child component: **Childapp.js**

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

//Child component
export default class Childapp extends Component {
    render() {
        return (
            <div>
                <p>Child Component</p>
            {/* prop "text" in Child component to which values will be passed
            from Parent component */}
                <p>{this.props.text}</p>
            </div>
        )
    }
}
```

Parent component: **Parentapp.js.** In the below example we are importing the child component **Childapp.js**

JS Parentapp.js U

src > JS Parentapp.js > ...

```
import React, { Component } from 'react'
import Childapp from "./Childapp"

//Parent Component
export default class App extends Component {
  render() {
    return (
      <div>
          <p>Parent Component</p>

      {/* we have passed a prop "text" from the parent to the child.  */}
         <Childapp text={"1st value"}/>
         <Childapp text={"2nd value"}/>

      </div>
    )
  }
}

```

In the below example we are importing Parentapp.js in **index.js file** to render.

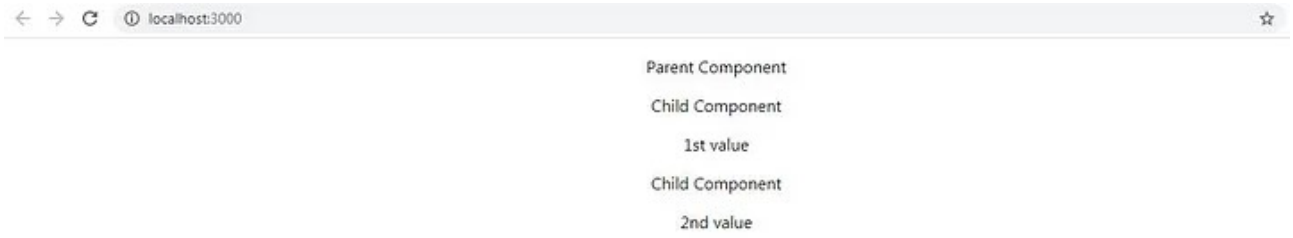
JS index.js M

src > JS index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import reportWebVitals from './reportWebVitals';
import Parentapp from "./Parentapp"

ReactDOM.render(
  <React.StrictMode>
    <Parentapp />
  </React.StrictMode>,
  document.getElementById('root')
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();

```

Index File

Output:



Example for ReactJS Props — #From Class to Functional Component

Passing data from Class component(Parent) to Functional component (Child) using props.

Below One Child component (a functional component)

Below One Parent component (a Class component)

JS Parentapp.js U

src > JS Parentapp.js > Parentapp > render

```
import React, { Component } from 'react'
import Childapp from "./Childapp"

//class component
export default class Parentapp extends Component {
    render() {

        return (
        <div>
        <Childapp greetings={"Hello Sandy"} quote={"Morning quote"}
        proverb={"evening proverb"} />
        </div>
        )
}
}
```

Parent component (Class component)

Output: