14. Create a simple function component (say Greeting) that takes a name prop and displays "Hello, [name]!".

//App.jsx

```
import React from "react";

const Greeting = (props) => {
  return <h1>Hello, {props.name}!</h1>;
};

const App = () => {
  return (
    <div>
      <Greeting name="John" />
    </div>
  );
};

export default App;
```

```
//main.jsx
import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import App from './App.jsx'


createRoot(document.getElementById('root')).render(
  <StrictMode>
    <App />
  </StrictMode>,
)
```

//index.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>Function Component</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

15. Create a simple class component (say Greeting) that displays "Hello, [name]!".

```
//Greeting.jsx
import React, { Component } from "react";

class Greeting extends Component {
  render() {
    return <h1>Hello, {this.props.name}!</h1>;
  }
}

export default Greeting;


//App.jsx
import Greeting from "./Greeting";

const App = () => {
  return (
    <div>
      <Greeting name="John" />
    </div>
  );
};

export default App;

//main.jsx
import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import App from './App.jsx'


createRoot(document.getElementById('root')).render(
  <StrictMode>
    <App />
  </StrictMode>
)
```

```
//index.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>Class Component</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

16. Program to illustrate how data can be passed from a parent component to a child component using **props**.

```
// ChildComponent.jsx
import React from 'react';

function ChildComponent(props) {
        // Accessing the data passed via props
        const { name, age, profession } = props.user;

        return (
             <div>
                    <h2>User Profile</h2>
                    <p>Name: {name}</p>
                    <p>Age: {age}</p>
                    <p>Profession: {profession}</p>
                </div>
        );
      }

export default ChildComponent;
```

```
// ParentComponent.jsx
import React from 'react';
import ChildComponent from './ChildComponent';
```

```
function ParentComponent() {
          const userData = {
               name: 'John Doe',
                       age: 28,
               profession: 'Software Developer'
                 };

        return (
            <div>
                <h1>Welcome to the User Profile</h1>
                {/* Passing userData as props to the ChildComponent */}
                <ChildComponent user={userData} />
         </div>
     );
          }

export default ParentComponent;
```

//main.jsx

```
import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import ParentComponent from './ParentComponent.jsx'
import ChildComponent from './ChildComponent.jsx';

const ele=createRoot(document.getElementById('root'))

ele.render(
  <div>
    <ParentComponent />

 </div>
)
```

```
//index.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
     <title> Passing Data via Props</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

17. Create a counter that increments when a button is clicked.

//App.jsx

```
import React, { useState } from "react";

const Counter = () => {
  const [count, setCount] = useState(0);

  const handleIncrement = () => {
    setCount(count + 1);
  };

  return (
    <div>
      <h1>Counter: {count}</h1>
      <button onClick={handleIncrement}>Increment</button>
    </div>
  );
};

const App = () => {
  return (
    <div>
      <Counter />
    </div>
  );
};

export default App;
```

```
//main.jsx
import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import App from './App.jsx'


createRoot(document.getElementById('root')).render(
  <StrictMode>
    <App />
  </StrictMode>,
)
```

//index.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
    <title>Event Handling</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```