

React

By Fred Bolder

Conditional rendering

EXAMPLE 1

ShowOnFriday.jsx

```
export default function ShowOnFriday() {  
  const today = new Date();  
  return today.getDay() === 5 && <div>It is almost weekend!</div>;  
}
```

App.jsx

```
import ShowOnFriday from "../components/ShowOnFriday";
```

```
export default function App() {  
  return (  
    <>  
      <div>Conditional rendering</div>  
      <ShowOnFriday />  
    </>  
  );  
}
```

EXAMPLE 2

YesNo.jsx

```
export default function YesNo(props) {  
  return <div>{props.condition ? "Yes" : "No"}</div>;  
}
```

App.jsx

```
import YesNo from "../components/YesNo";
```

```
export default function App() {  
  return (  
    <>  
      <div>Conditional rendering</div>  
      <YesNo condition={true} />  
      <YesNo condition={false} />  
    </>  
  );  
}
```

Installing

Example 1

```
npm create vite@latest  
Project name: test  
Select React  
Select JavaScript + SWC  
cd test  
npm install  
npm run dev
```

Example 2

```
npx create-react-app test  
cd test  
npm start
```

React Router

Example 1

Terminal

```
npx create-react-app example
cd example
npm i react-router-dom
npm start
```

App.jsx

```
import { BrowserRouter, Routes, Route, Link, Outlet } from "react-router-dom";
import Home from "../components/Home";
import Links from "../components/Links";
import PageNotFound from "../components/PageNotFound";
```

```
export default function App() {
  return (
    <div>
      <BrowserRouter>
        <div
          style={{
            display: "flex",
            justifyContent: "center",
            columnGap: "1em",
          }}
        >
          <Link to="/">Home</Link>
          <Link to="/links">Links</Link>
        </div>
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/links" element={<Links />} />
          <Route path="*" element={<PageNotFound />} />
        </Routes>
      </BrowserRouter>
    </div>
  );
}
```

Home.jsx

```
export default function Home() {
  return (
    <div>
      <h1>Home</h1>
      <p>This is the Home page</p>
    </div>
  );
}
```

Links.jsx

```
export default function Links() {
  return (
    <div>
      <h1>Links</h1>
      <a
        href="https://www.google.com"
        target="_blank"
        rel="noopener noreferrer"
      >
        Google
      </a>
    </div>
  );
}
```

PageNotFound.jsx

```
export default function PageNotFound() {
  return (
    <div>
      <h1>Page not found</h1>
    </div>
  );
}
```

Example 2

Terminal

```
npx create-react-app example
cd example
npm i react-router-dom
npm start
```

index.js

```
import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App";
import { BrowserRouter } from "react-router-dom";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>
);
```

App.jsx

```
import { Routes, Route, useNavigate } from "react-router-dom";
import Home from "../components/Home";
import PageNotFound from "../components/PageNotFound";
import Product from "../components/Product";

export default function App() {
  const navigate = useNavigate();

  function handleClick(e) {
    navigate(e.target.value === "home" ? "/" : "/product/" + e.target.value);
  }

  return (
    <div>
      <div
        style={{
          display: "flex",
          justifyContent: "center",
          columnGap: "1em",
          margin: "1em",
        }}
      >
        <button value="home" onClick={handleClick}>
          Home
        </button>
        <button value="1" onClick={handleClick}>
          Product 1
        </button>
        <button value="2" onClick={handleClick}>
          Product 2
        </button>
      </div>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/product/:ID" element={<Product />} />
        <Route path="*" element={<PageNotFound />} />
      </Routes>
    </div>
  );
}
```

Home.jsx

```
export default function Home() {
  return (
    <div>
      <h1>Home</h1>
      <p>This is the Home page</p>
    </div>
  );
}
```

Product.jsx

```
import { useParams } from "react-router-dom";
```

```
export default function Product() {  
  return (  
    <>  
      <h1>{`Product ${useParams().ID}`}</h1>  
    </>  
  );  
}
```

PageNotFound.jsx

```
export default function PageNotFound() {  
  return (  
    <div>  
      <h1>Page not found</h1>  
    </div>  
  );  
}
```

useCallback

<https://react.dev/reference/react/useCallback>

useContext

App.jsx

```
import './App.css';
import Header from './components/Header';
import Content from './components/Content';
import GlobalState from './context/GlobalState';
```

```
function App() {
  return (
    <>
      <GlobalState>
        <Header />
        <Content />
      </GlobalState>
    </>
  );
}
```

```
export default App;
```

LanguageContext.jsx

```
import { createContext } from "react";
```

```
const LanguageContext = createContext();
```

```
export default LanguageContext;
```

GlobalState.jsx

```
import { useState } from "react";
import LanguageContext from './LanguageContext';
```

```
function GlobalState(props) {
  const [lang, setLang] = useState("ENG");
  return (
    <LanguageContext.Provider value={{ lang, setLang }}>
      {props.children}
    </LanguageContext.Provider>
  );
}
```

```
export default GlobalState;
```

Header.jsx

```
import LanguageChooser from './LanguageChooser';
```

```
export default function Header() {
  return (
    <div>
      <LanguageChooser />
    </div>
  );
}
```

LanguageChooser.jsx

```
import { useContext } from "react";
import LanguageContext from './context/LanguageContext';
```

```

const LanguageChooser = () => {
  const { setLang } = useContext(LanguageContext);

  return (
    <div className="select">
      <select
        id="selectLanguage"
        placeholder="test"
        onChange = {(e) => {
          setLang(e.target.value);
        }}
      >
        <option value="ENG">English</option>
        <option value="GER">German</option>
        <option value="DUT">Dutch</option>
      </select>
    </div>
  );
};

export default LanguageChooser;

```

WelcomeMessage.jsx

```

import { useContext } from "react";
import LanguageContext from "../context/LanguageContext";

export default function WelcomeMessage() {
  const { lang } = useContext(LanguageContext);
  let msg = "";
  switch (lang) {
    case "ENG":
      msg = "Hello World!";
      break;
    case "GER":
      msg = "Hallo Welt!";
      break;
    case "DUT":
      msg = "Hallo wereld!";
      break;
    default:
      msg = "";
      break;
  }
  return (
    <div>
      <h1>{msg}</h1>
    </div>
  );
}

```

Content.jsx

```
import WelcomeMessage from "./WelcomeMessage";
```

```
export default function Content() {
```

```
  return (
```

```
    <div>
```

```
      <WelcomeMessage />
```

```
    </div>
```

```
  );
```

```
}
```

useEffect

```
import { useEffect, useState } from 'react';

function MyComponent(props) {
  const [state, setState] = useState("");

  useEffect(() => {
    // Runs after EVERY rendering
  });

  useEffect(() => {
    // Runs ONCE after initial rendering
  }, []);

  useEffect(() => {
    // Runs ONCE after initial rendering
    // and after every rendering ONLY IF `props` or `state` changes
  }, [props, state]);
}
```

<https://dmitripavlutin.com/react-useeffect-explanation/>
<https://dev.to/arikaturika/the-traps-of-useeffect-infinite-loops-836>

useReducer

```
import './App.css';
import { useReducer } from 'react';

const countReducer = (state, action) => {
  if (action.type === 'increase') {
    return state + 1;
  }
  if (action.type === 'set') {
    return action.value;
  }
}

function App() {
  const [count, dispatch] = useReducer(countReducer, 0);
  return (
    <div className="App">
      <h1>Count: {count}</h1>
      <button onClick={() => dispatch({ type: 'increase' })}>Increase</button>
      <button onClick={() => dispatch({ type: 'set', value: 5 })}>
        Set to value
      </button>
    </div>
  );
}

export default App;
```

<https://react.dev/reference/react/useReducer>

useState

Example 1

Counter.jsx

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

export default function Counter() {
  const [count, setCount] = useState(0);

  function increase() {
    setCount(count + 1);
  }

  // useState renders after change
  return <h1 onClick={increase}>{count.toString()}</h1>;
}
```

App.jsx

```
import Counter from "../components/Counter";

export default function App() {
  return (
    <>
      <div>Click on the number to increase it</div>
      <Counter />
    </>
  );
}
```

Example 2

App.jsx

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

export default function App() {
  const [ text, setText ] = useState("");

  const handleChange = (event) => {
    setText(event.target.value);
  }

  return (
    <>
      <input type="text" value = {text} onChange={handleChange} />
      <div>{text.toUpperCase()}</div>
    </>
  );
}
```