**Conditional Rendering**

As we can understand from the term, conditional rendering is a way to render different JSX or component at different condition. We can implement conditional rendering using regular if and else statement, ternary operator and &&. Let's implement a different conditional rendering.

**Conditional Rendering using If and Else statement**

// index.js

import React from 'react'

import ReactDOM from 'react-dom'

// class based component

class Header extends React.Component {

render() {

const {

welcome,

title,

subtitle,

author: { firstName, lastName },

date,

} = this.props.data

return (

<header>

<div className='header-wrapper'>

<h1>{welcome}</h1>

<h2>{title}</h2>

<h3>{subtitle}</h3>

<p>

{firstName} {lastName}

</p>

<small>{date}</small>

<p>Select a country for your next holiday</p>

</div>

</header>

)

}

}

class App extends React.Component {

state = {

loggedIn: false,

}

render() {

const data = {

welcome: '30 Days Of React',

title: 'Getting Started React',

subtitle: 'JavaScript Library',

author: {

firstName: 'Asabeneh',

lastName: 'Yetayeh',

},

date: 'Oct 9, 2020',

}

// conditional rendering using if and else statement

let status

if (this.state.loggedIn) {

status = <h3>Welcome to 30 Days Of React</h3>

} else {

status = <h3>Please Login</h3>

}

return (

<div className='app'>

<Header data={data} />

{status}

</div>

)

}

}

const rootElement = document.getElementById('root')

ReactDOM.render(<App />, rootElement)

Let's add a method which allow as to toggle the status of the user. We should have a button to handle event for logging in and logging out.

// index.js

import React from 'react'

import ReactDOM from 'react-dom'

// A button component

const Button = ({ text, onClick, style }) => (

<button style={style} onClick={onClick}>

{text}

</button>

)

// CSS styles in JavaScript Object

const buttonStyles = {

backgroundColor: '#61dbfb',

padding: 10,

border: 'none',

borderRadius: 5,

margin: '3px auto',

cursor: 'pointer',

fontSize: 22,

color: 'white',

}

// class based component

class Header extends React.Component {

render() {

console.log(this.props.data)

const {

welcome,

title,

subtitle,

author: { firstName, lastName },

date,

} = this.props.data

return (

<header>

<div className='header-wrapper'>

<h1>{welcome}</h1>

<h2>{title}</h2>

<h3>{subtitle}</h3>

<p>

{firstName} {lastName}

</p>

<small>{date}</small>

</div>

</header>

)

}

}

class App extends React.Component {

state = {

loggedIn: false,

}

handleLogin = () => {

this.setState({

loggedIn: !this.state.loggedIn,

})

}

render() {

const data = {

welcome: '30 Days Of React',

title: 'Getting Started React',

subtitle: 'JavaScript Library',

author: {

firstName: 'Asabeneh',

lastName: 'Yetayeh',

},

date: 'Oct 9, 2020',

}

let status

let text

if (this.state.loggedIn) {

status = <h1>Welcome to 30 Days Of React</h1>

text = 'Logout'

} else {

status = <h3>Please Login</h3>

text = 'Login'

}

return (

<div className='app'>

<Header data={data} />

{status}

<Button text={text} style={buttonStyles} onClick={this.handleLogin} />

</div>

)

}

}

const rootElement = document.getElementById('root')

ReactDOM.render(<App />, rootElement)

How about if our condition is more than two? Like pure JavaScript we can use if else if statement. In general, conditional rendering is not different from pure JavaScript conditional statement.

**Conditional Rendering using Ternary Operator**

Ternary operator is an an alternative for if else statement. However, there is more use cases for ternary operator than if else statement. For example, use can use ternary operator inside styles, className or many places in a component than regular if else statement.

// index.js

import React from 'react'

import ReactDOM from 'react-dom'

// A button component

const Button = ({ text, onClick, style }) => (

<button style={style} onClick={onClick}>

{text}

</button>

)

// CSS styles in JavaScript Object

const buttonStyles = {

backgroundColor: '#61dbfb',

padding: 10,

border: 'none',

borderRadius: 5,

margin: '3px auto',

cursor: 'pointer',

fontSize: 22,

color: 'white',

}

// class based component

class Header extends React.Component {

render() {

const {

welcome,

title,

subtitle,

author: { firstName, lastName },

date,

} = this.props.data

return (

<header>

<div className='header-wrapper'>

<h1>{welcome}</h1>

<h2>{title}</h2>

<h3>{subtitle}</h3>

<p>

{firstName} {lastName}

</p>

<small>{date}</small>

</div>

</header>

)

}

}

class App extends React.Component {

state = {

loggedIn: false,

}

handleLogin = () => {

this.setState({

loggedIn: !this.state.loggedIn,

})

}

render() {

const data = {

welcome: '30 Days Of React',

title: 'Getting Started React',

subtitle: 'JavaScript Library',

author: {

firstName: 'Asabeneh',

lastName: 'Yetayeh',

},

date: 'Oct 9, 2020',

}

let status = this.state.loggedIn ? (

<h1>Welcome to 30 Days Of React</h1>

) : (

<h3>Please Login</h3>

)

return (

<div className='app'>

<Header data={data} />

{status}

<Button

text={this.state.loggedIn ? 'Logout' : 'Login'}

style={buttonStyles}

onClick={this.handleLogin}

/>

</div>

)

}

}

const rootElement = document.getElementById('root')

ReactDOM.render(<App />, rootElement)

In addition to JSX, we can also conditionally render a component. Let's change the above conditional JSX to a component.

// index.js

import React from 'react'

import ReactDOM from 'react-dom'

// A button component

const Button = ({ text, onClick, style }) => (

<button style={style} onClick={onClick}>

{text}

</button>

)

// CSS styles in JavaScript Object

const buttonStyles = {

backgroundColor: '#61dbfb',

padding: 10,

border: 'none',

borderRadius: 5,

margin: '3px auto',

cursor: 'pointer',

fontSize: 22,

color: 'white',

}

// class based component

class Header extends React.Component {

render() {

const {

welcome,

title,

subtitle,

author: { firstName, lastName },

date,

} = this.props.data

return (

<header>

<div className='header-wrapper'>

<h1>{welcome}</h1>

<h2>{title}</h2>

<h3>{subtitle}</h3>

<p>

{firstName} {lastName}

</p>

<small>{date}</small>

</div>

</header>

)

}

}

const Login = () => (

<div>

<h3>Please Login</h3>

</div>

)

const Welcome = (props) => (

<div>

<h1>Welcome to 30 Days Of React</h1>

</div>

)

class App extends React.Component {

state = {

loggedIn: false,

}

handleLogin = () => {

this.setState({

loggedIn: !this.state.loggedIn,

})

}

render() {

const data = {

welcome: '30 Days Of React',

title: 'Getting Started React',

subtitle: 'JavaScript Library',

author: {

firstName: 'Asabeneh',

lastName: 'Yetayeh',

},

date: 'Oct 9, 2020',

}

const status = this.state.loggedIn ? <Welcome /> : <Login />

return (

<div className='app'>

<Header data={data} />

{status}

<Button

text={this.state.loggedIn ? 'Logout' : 'Login'}

style={buttonStyles}

onClick={this.handleLogin}

/>

</div>

)

}

}

const rootElement = document.getElementById('root')

ReactDOM.render(<App />, rootElement)

**Conditional Rendering using && Operator**

The && operator render the right JSX operand if the left operand(expression) is true.

// index.js

import React from 'react'

import ReactDOM from 'react-dom'

// A button component

const Button = ({ text, onClick, style }) => (

<button style={style} onClick={onClick}>

{text}

</button>

)

// CSS styles in JavaScript Object

const buttonStyles = {

backgroundColor: '#61dbfb',

padding: 10,

border: 'none',

borderRadius: 5,

margin: '3px auto',

cursor: 'pointer',

fontSize: 22,

color: 'white',

}

// class based component

class Header extends React.Component {

render() {

console.log(this.props.data)

const {

welcome,

title,

subtitle,

author: { firstName, lastName },

date,

} = this.props.data

return (

<header style={this.props.styles}>

<div className='header-wrapper'>

<h1>{welcome}</h1>

<h2>{title}</h2>

<h3>{subtitle}</h3>

<p>

{firstName} {lastName}

</p>

<small>{date}</small>

</div>

</header>

)

}

}

const Login = () => (

<div>

<h3>Please Login</h3>

</div>

)

const Welcome = (props) => (

<div>

<h1>Welcome to 30 Days Of React</h1>

</div>

)

class App extends React.Component {

state = {

loggedIn: false,

techs: ['HTML', 'CSS', 'JS'],

}

handleLogin = () => {

this.setState({

loggedIn: !this.state.loggedIn,

})

}

render() {

const data = {

welcome: '30 Days Of React',

title: 'Getting Started React',

subtitle: 'JavaScript Library',

author: {

firstName: 'Asabeneh',

lastName: 'Yetayeh',

},

date: 'Oct 9, 2020',

}

// We can destructure state

const { loggedIn, techs } = this.state

const status = loggedIn ? <Welcome /> : <Login />

return (

<div className='app'>

<Header data={data} />

{status}

<Button

text={loggedIn ? 'Logout' : 'Login'}

style={buttonStyles}

onClick={this.handleLogin}

/>

{techs.length === 3 && (

<p>You have all the prerequisite courses to get started React</p>

)}

{!loggedIn && (

<p>

Please login to access more information about 30 Days Of React

challenge

</p>

)}

</div>

)

}

}

const rootElement = document.getElementById('root')

ReactDOM.render(<App />, rootElement)

In the previous section, we used alert box to greet people and also to display time. Let's render the greeting and time on browser DOM instead of displaying on alert box.

// index.js

import React from 'react'

import ReactDOM from 'react-dom'

// class based component

class Header extends React.Component {

render() {

console.log(this.props.data)

const {

welcome,

title,

subtitle,

author: { firstName, lastName },

date,

} = this.props.data

return (

<header style={this.props.styles}>

<div className='header-wrapper'>

<h1>{welcome}</h1>

<h2>{title}</h2>

<h3>{subtitle}</h3>

<p>

{firstName} {lastName}

</p>

<small>{date}</small>

</div>

</header>

)

}

}

const Message = ({ message }) => (

<div>

<h1>{message}</h1>

</div>

)

const Login = () => (

<div>

<h3>Please Login</h3>

</div>

)

const Welcome = (props) => (

<div>

<h1>Welcome to 30 Days Of React</h1>

</div>

)

// A button component

const Button = ({ text, onClick, style }) => (

<button style={style} onClick={onClick}>

{text}

</button>

)

// TechList Component

// class base component

class TechList extends React.Component {

render() {

const { techs } = this.props

const techsFormatted = techs.map((tech) => <li key={tech}>{tech}</li>)

return techsFormatted

}

}

// Main Component

// Class Component

class Main extends React.Component {

render() {

const {

techs,

greetPeople,

handleTime,

loggedIn,

handleLogin,

message,

} = this.props

console.log(message)

const status = loggedIn ? <Welcome /> : <Login />

return (

<main>

<div className='main-wrapper'>

<p>Prerequisite to get started react.js:</p>

<ul>

<TechList techs={this.props.techs} />

</ul>

{techs.length === 3 && (

<p>You have all the prerequisite courses to get started React</p>

)}

<div>

<Button

text='Show Time'

onClick={handleTime}

style={buttonStyles}

/>{' '}

<Button

text='Greet People'

onClick={greetPeople}

style={buttonStyles}

/>

{!loggedIn && <p>Please login to access more information about 30 Days Of React challenge</p>}

</div>

<div style={{ margin: 30 }}>

<Button

text={loggedIn ? 'Logout' : 'Login'}

style={buttonStyles}

onClick={handleLogin}

/>

<br />

{status}

</div>

<Message message={message} />

</div>

</main>

)

}

}

// CSS styles in JavaScript Object

const buttonStyles = {

backgroundColor: '#61dbfb',

padding: 10,

border: 'none',

borderRadius: 5,

margin: '3px auto',

cursor: 'pointer',

fontSize: 22,

color: 'white',

}

// Footer Component

// Class component

class Footer extends React.Component {

constructor(props) {

super(props)

}

render() {

return (

<footer>

<div className='footer-wrapper'>

<p>Copyright {this.props.date.getFullYear()}</p>

</div>

</footer>

)

}

}

class App extends React.Component {

state = {

loggedIn: false,

techs: ['HTML', 'CSS', 'JS'],

message: 'Click show time or Greet people to change me',

}

handleLogin = () => {

this.setState({

loggedIn: !this.state.loggedIn,

})

}

showDate = (time) => {

const months = [

'January',

'February',

'March',

'April',

'May',

'June',

'July',

'August',

'September',

'October',

'November',

'December',

]

const month = months[time.getMonth()].slice(0, 3)

const year = time.getFullYear()

const date = time.getDate()

return `${month} ${date}, ${year}`

}

handleTime = () => {

let message = this.showDate(new Date())

this.setState({ message })

}

greetPeople = () => {

let message = 'Welcome to 30 Days Of React Challenge, 2020'

this.setState({ message })

}

render() {

const data = {

welcome: '30 Days Of React',

title: 'Getting Started React',

subtitle: 'JavaScript Library',

author: {

firstName: 'Asabeneh',

lastName: 'Yetayeh',

},

date: 'Oct 9, 2020',

}

return (

<div className='app'>

<Header data={data} />

<Main

techs={techs}

handleTime={this.handleTime}

greetPeople={this.greetPeople}

loggedIn={this.state.loggedIn}

handleLogin={this.handleLogin}

message={this.state.message}

/>

<Footer date={new Date()} />

</div>

)

}

}

const rootElement = document.getElementById('root')

ReactDOM.render(<App />, rootElement)