React.5

patterns

https://reactpatterns.js.org/docs/

▼ What is a proxy component?

A proxy component is a placeholder component that can be rendered to or from another component.

In short a proxy component is a **reusable component**.

```
import React from 'react'

class Button extends React.Component {
  render() {
    return <button type="button">My Button</button>
  }
}

class App extends React.Component {
  render() {
    return <Button />
  }
}

export default App
```

▼ Make the API Call in the useEffect hook

you can use setState to update your component when the data is retrieved

```
useEffect(() => {
  fetch('api/sms')
    .then(result => {
      const sms = result.data
      console.log('COMPONENT WILL Mount messages : ', sms)
      this.setState({sms: [...sms.content]})
    })
}, [])
```

▼ What is a stateless function?

a way to define React **presentational** components as a function

Stateless function does not hold state; just props

```
const UserPassword = function(props) {
  return <p>The user password is: {this.props.userpassword}</p>
};
```

▼ What is a Higher-Order Function?

Functions that operate on other functions, either by taking them as arguments or by returning them are called higher-order functions.

```
function unless(test, then) {
  if (!test) then()
}

repeat(3, n => {
  unless(n % 2 == 1, () => {
    console.log(n, "is even")
  })
})
// 0 is even
// 2 is even
```

▼ What is a Higher-Order Component?

A higher-order component is a function that takes a component and returns a new component.

```
const withColor = Element => props => <Element {...props} color="red" />

const Button = () => {
  return <button>My Button</button>
}

const ColoredButton = withColor(Button)
```

▼ How to access a child's DOM node from a parent component?

through Refs

```
import React from 'react'

class Input extends React.Component {
  constructor(props) {
    super(props)
    // create a ref to store the textInput DOM element
    this.textInput = React.createRef()
  }

  focus() {
    // EXPLANATION: a reference to the node becomes accessible at the current attribute of the ref.
    // make the DOM node focus
    this.textInput.current.focus();
  }

  render() {
    return (
      <input
        type="text"
        ref={this.textInput}
      />
    )
  }
}
```

- What are JSX spread attributes?

  it's a syntax for passing all of an object's properties as JSX attributes

```
const component = <Component {...props} /
```

- What is a render callback?

  For example below, notice that we create a function foo which takes a callback function as a parameter.

  When we call foo, it turns around and calls back to the passed-in function.

```
const foo = (hello) => {
  return hello('foo')
}

foo((name) => {
  return `hello from ${name}`
})
```

- What is Function as a Child Component?

  A Function as child component is a pattern that lets you pass a render function to a component as the children prop so **you can change what you can pass as children to a component**.

```
import React from 'react'

class PageWidth extends React.Component {
  state = { width: 0 }

  componentDidMount() {
    this.setState({ width: window.innerWidth })

    window.addEventListener(
      'resize',
      ({ target }) => {
        this.setState({ width: target.innerWidth })
      }
    )
  }

  render() {
    const { width } = this.state

    return this.props.children(width)
  }
}

<PageWidth>
  {width => <div>Page width is {width}</div>}
</PageWidth>
```

▼ What is Function as a Prop Component?

this is how you can pass a function as a prop to a component

```
const hello = (name) => {
  return <div>`hello from ${name}`</div>
}

const Foo = ({ hello }) => {
  return hello('foo')
}

<Foo hello={hello} />
```

▼

```
```

▼

```
```

▼

▼